

AD-A069 029

TRW DEFENSE AND SPACE SYSTEMS GROUP REDONDO BEACH CALIF  
LAMAS SYSTEM MANUAL. (U)  
FEB 78

F/6 9/2

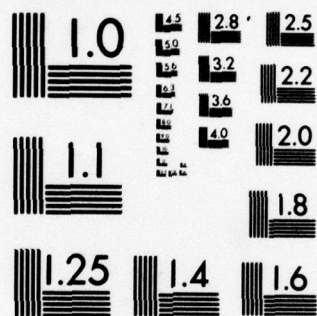
DAAG39-77-C-0112

NL

UNCLASSIFIED

1 OF 2  
AD  
A069 029





MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A



DDC FILE COPY

LEVEL

AD A069029

ADVANCED STUDIES

ONE SPACE PARK - REDONDO BEACH - CALIFORNIA 90278

# LAMAS SYSTEM MANUAL

COML ITEM A008

FEBRUARY 1978

DDC  
RECEIVED  
MAR 23 1978  
RUSSELL C

CONTRACT NO. DAAG39-77-C-0112

This document has been approved  
for public release and sale; its  
distribution is unlimited.

Approved for:  
Training Component-Arms Test Activity  
U.S. Army Research Center and School

79 04 09 147

## **DISCLAIMER NOTICE**

**THIS DOCUMENT IS BEST QUALITY  
PRACTICABLE. THE COPY FURNISHED  
TO DDC CONTAINED A SIGNIFICANT  
NUMBER OF PAGES WHICH DO NOT  
REPRODUCE LEGIBLY.**



**TRW.**

DEFENSE AND SPACE SYSTEMS GROUP

**ADVANCED STUDIES**

ONE SPACE PARK • REDONDO BEACH • CALIFORNIA 90278

**LAMAS SYSTEM MANUAL.**

CDRL ITEM A008

FEB 1978

139p.

CONTRACT NO DAAG39-77-C-1112

DDC  
RECEIVED  
MAY 23 1978  
ALBUQUERQUE

Prepared for:

TRADOC Combined-Arms Test Activity  
U.S.A. Intelligence Center and School

This document has been approved  
for public release and sale; its  
distribution is unlimited.

409637

79 04 09 147

## LAMAS SYSTEM MANUAL

This document describes the LAMAS software package from both a programmer and user standpoint. The text is written in Program Design Language (PDL) format and represents the final version of the working document used throughout program development. All LAMAS software and manual procedures are described, with the exception of the display software developed using company funds. A proprietary version of this document which describes both deliverable and company-funded software is maintained on file in the TRW Advanced Studies office.

ACCESSION for	White Section <input checked="" type="checkbox"/>	Buff Section <input type="checkbox"/>
NTIS	by <i>for file</i>	
DOC	UNANNOUNCED	
JUSTIFICATION	BY <i>for file</i>	
DISTRIBUTION/AVAILABILITY CODES		
SPECIAL		
A 23		



TRW SYSTEMS GROUP  
ONE SPACE PARK  
REDONDI BEACH, CALIFORNIA 90278  
(213) 535-4321

\*\*\*\*\*  
\* LOCATION AND MOVEMENT ANALYSIS SYSTEM \*  
\* (LAMAS) \*  
\* SOFTWARE TO COMPUTE GROUND FORCE MOVEMENT \*  
\* CONSIDERING MOVEMENT DOCTRINE, ROAD CONDITIONS, \*  
\* AND RISK FACTORS \*  
\* 21 FEB 78 \*  
\* PDL 03.06 \*  
\*\*\*\*\*

## TABLE OF CONTENTS

GENERAL DESCRIPTION . . . . .	2
OVERVIEW . . . . .	3
INITIALIZATION . . . . .	4
GROUND MOVEMENT CALCULATIONS . . . . .	5
USER FORMAT . . . . .	6
SYSTEM ARCHITECTURE . . . . .	7
DATA BASE DESCRIPTIONS . . . . .	8
NODES AND LINKS . . . . .	9
NODE VECTOR FORMAT . . . . .	10
NODE VECTOR DESCRIPTION . . . . .	11
TERRAIN . . . . .	12
CROSS-COUNTRY AND CONCEALMENT CODE DESCRIPTIONS . . . . .	13
UNITS . . . . .	14
UNIT VECTOR FORMAT . . . . .	15
UNIT VECTOR DESCRIPTION . . . . .	16
SOLUTIONS . . . . .	17
SOLUTION VECTOR FORMAT . . . . .	18
SOLUTION VECTOR DESCRIPTION . . . . .	19
ROUTES . . . . .	20
ROUTE VECTOR FORMAT . . . . .	21
ROUTE VECTOR DESCRIPTION . . . . .	22
MAP DIRECTORY . . . . .	23
CROSS-COUNTRY DIRECTORY . . . . .	24
PREPARATION FOR USE . . . . .	25
CREATE NODE VECTOR AND MAP DIRECTORY FILES ON DISK . . . . .	26
CREATE NODE CARD . . . . .	27
NODE CARD FORMAT . . . . .	28
CREATE LINK CARD . . . . .	29
LINK CARD FORMAT . . . . .	30
CREATE MAP DIRECTORY ENTRY FOR THIS MAP . . . . .	31
CREATE CROSS-COUNTRY, CONCEALMENT, AND DIRECTORY FILES ON DISK . . . . .	32
LAWAS FLOW . . . . .	33
LAWAS FLOW . . . . .	34



TABLE OF CONTENTS

GENERAL DESCRIPTION . . . . .	2
OVERVIEW . . . . .	3
INITIALIZATION . . . . .	4
GROUND MOVEMENT CALCULATIONS . . . . .	5
USER FORMAT . . . . .	6
SYSTEM ARCHITECTURE . . . . .	7
DATA BASE DESCRIPTIONS . . . . .	8
NODES AND LINKS . . . . .	9
NODE VECTOR FORMAT . . . . .	10
NODE VECTOR DESCRIPTION . . . . .	11
TERRAIN . . . . .	12
CROSS-COUNTRY AND CONCEALMENT CODE DESCRIPTIONS . . . . .	13
UNITS . . . . .	14
UNIT+VECTOR FORMAT . . . . .	15
UNIT+VECTOR DESCRIPTION . . . . .	16
SOLUTIONS . . . . .	17
SOLUTION+VECTOR FORMAT . . . . .	18
SOLUTION+VECTOR DESCRIPTION . . . . .	19
ROUTES . . . . .	20
ROUTE+VECTOR FORMAT . . . . .	21
ROUTE+VECTOR FORMAT DESCRIPTION . . . . .	22
MAP DIRECTORY . . . . .	23
CROSS-COUNTRY DIRECTORY . . . . .	24
PREPARATION FOR USE . . . . .	25
PREPARATION . . . . .	26
CREATE NODE+VECTOR AND MAP+DIRECTORY FILES ON DISK . . . . .	27
CREATE NODE CARD . . . . .	28
NODE CARD FORMAT . . . . .	29
CREATE LINK CARD . . . . .	30
LINK CARD FORMAT . . . . .	31
CREATE MAP+DIRECTORY ENTRY FOR THIS MAP . . . . .	32
CREATE CROSS-COUNTRY, CONCEALMENT, AND DIRECTORY FILES ON DISK . . . . .	33
LAHAS FLOW . . . . .	34

LAMAS	35
IMPLEMENT ALGORITHMS AND FUNCTIONS USING THE NODE+VECTOR DATA BASE	36
PREPARE FOR ROAD NETWORK PATH CALCULATIONS	37
INITIALIZE MAP+NUMBERS AND SCREEN EXTREMES	38
ESTABLISH DIRECTORY IN MEMORY AND READ IN NODE+VECTORS	39
READ THIS MAP'S NODES INTO MAIN MEMORY	40
CHANGE ADJACENT NODE'S MAP+NUMBER TO INDEX	41
INITIALIZE UNIT+VECTORS	42
ESTABLISH A FORWARD EDGE OF BATTLE AREA	43
FIND NODE+VECTOR NEAREST TO UTM COORDINATES	44
PATH DETERMINATION AND DISPLAY	45
PERFORM INTERACTIVE OPERATIONS	46
PRINT MAP+NUMBER, NODE+NUMBER OF NODE+VECTOR NEAREST GIVEN COORDINATES	47
CHANGE CONTENTS OF A NODE+VECTOR	48
CHANGE CHARACTERISTIC	49
DELETE A NODE FROM THE NODE+VECTOR ARRAY	50
PRINT CONTENTS OF A NODE+VECTOR	51
CHANGE PROPERTIES OF A UNIT	52
CALCULATE PATH	53
PURGE ALL ROUTES	54
REINITIALIZE 'WORKING LIST' NODE+VECTOR ENTRIES	55
CALCULATE AND ASSIGN TIME AND NORTH VALUES	56
COMPUTE NEXT NODE TO LABEL	57
EXPLANATION OF PATH DETERMINATION ALGORITHM	58
COMPUTE SOLUTION+VECTORS AND ROUTE+VECTOR	59
CALCULATE SECOND BEST PATH	60
FIND THE ORDER OF PRIORITIES FOR BEST NETWORK TRAVEL	61
CALCULATE BEST NODE AT WHICH TO INTERDICT	62
PRESENT RESULTS	63
PRINT TABLE OF ROUTE NUMBERS WITH ASSOCIATED UNIT NAMES	64
PRINT PATH STATISTICS	65
IMPLEMENT ALGORITHMS AND FUNCTIONS USING THE CROSS-COUNTRY DATA BASE	66
PERFORM TERRAIN INITIALIZATION	67
MAP+NUMBER TO BE USED	68
ESTABLISH UNIT+VECTORS	69
CHANGE ENTERED COORDINATES TO AN INDEX INTO THE DATA BASE	70
PERFORM TERRAIN PATH CALCULATIONS	71
CALCULATE CROSS-COUNTRY PATH	72
OBTAIN UNIT NAMES TO BE CONSIDERED	73
OBTAIN THE PATH PARAMETERS	74
CALCULATE THE CROSS-COUNTRY PATH	75



TRJ, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
TABLE OF CONTENTS

PAGE 1.003

CALCULATE THE ADJACENT NODE'S TIME VALUES . . . . .	76
CONSTRUCT SOLUTION+VECTORS AND ROUTE+VECTOR . . . . .	77
DISPLAY RESULTS . . . . .	78
DATA INDEX . . . . .	79
FLOW SEGMENT INDEX . . . . .	80
SEGMENT REFERENCE TREES . . . . .	81

TRU, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM

PAGE 2

\*\*\*\*\*  
\*  
\* GENERAL DESCRIPTION \*  
\*  
\*\*\*\*\*



OVERVIEW

```
#####  
# 1 THE PURPOSE OF THIS SOFTWARE PACKAGE IS TO COMPUTE GROUND  
# 2 FORCE MOVEMENT USING OPTIMAL PATH DETERMINATION AND ROUTE CONFLICT  
# 3 RESOLUTION ALGORITHMS. TO INSURE FLEXIBILITY, USER INTERACTION IS  
# 4 SUPPORTED DURING MOST PHASES OF THE SOFTWARE'S OPERATION.  
# 5 THREE MAJOR AREAS ARE INVOLVED IN THE PROCESS: INITIALIZATION  
# 6 GROUND FORCE MOVEMENT CALCULATIONS, AND DISPLAY.  
#  
#####  
#  
#####
```

10  
11  
12  
13  
14  
15

# INITIALIZATION

```
#####
# 1 INITIALIZATION IS THE PROCESS OF PREPARING DATA BASES FOR
# 2 GENERAL USE. THERE ARE THREE BASIC TYPES OF DATA BASES: PERMA-
# 3 NENT, TEMPORARY - USER DEFINED, AND TEMPORARY - PROGRAM DEFINED.
# 4 PERMANENT DATA BASES ARE CREATED BEFORE THE USER EVER
# 5 INVOKES THE SYSTEM, AND ARE GENERALLY UNALTERABLE BY THE USER.
# 6 INCLUDED ARE SUCH FILES AS NODES, ROADS, CITIES, CROSS-COUNTRY MOVE-
# 7 MENT, CONCEALMENT, MAP+DIRECTORY, ROAD+DIRECTORY, AND CROSS-COUNTRY
# 8 DIRECTORY.
# 9 A USER-DEFINED TEMPORARY DATA BASE IS A FILE OVER WHICH THE
# 10 USER HAS CONSIDERABLE CONTROL, BUT WHICH EXISTS ONLY FOR A PAR-
# 11 TICULAR SESSION ON THE SYSTEM. ONCE THE USER SIGNS OFF, THE DATA
# 12 BASE IS LOST. THE UNITS DATA BASE IS THE PRIMARY EXAMPLE OF THIS
# 13 TYPE IN LAMAS. HERE THE USER DEFINES THE CONTENTS, AND MAY ALTER
# 14 THE CONTENTS AT ANY TIME, BUT THE FILE DOES NOT CARRY OVER FROM
# 15 ONE SESSION TO ANOTHER.
# 16 PROGRAM-DEFINED TEMPORARY DATA BASES ARE FILES OVER WHICH
# 17 THE USER HAS NO CONTROL WHATSOEVER. THE FILES MAY ENABLE THE USER
# 18 TO DISPLAY SOME RESULTS, BUT NO ALTERATIONS MAY BE MADE. EX-
# 19 AMPLES OF THIS TYPE OF DATA BASE ARE SOLUTION+VECTORS, ROUTE+
# 20 VECTORS, MAP+DIRECTORY IN MEMORY, AND ROAD+DIRECTORY IN MEMORY.
# 21 THE EFFECT OF INITIALIZATION VARIES FROM FILE TO FILE.
# 22 FOR SOME, IT IS JUST SETTING ALL ELEMENTS TO ZERO, AS IN THE CASE
# 23 OF THE SOLUTION+VECTORS AND ROUTE+VECTORS. FOR OTHERS, IT IS THE
# 24 ESTABLISHING OF VALUES, AS IN ALL OF THE PERMANENT FILES, UNITS DATA
# 25 BASE, AND THE DIRECTORIES.
# 26 THE MANNER IN WHICH THE VALUES ARE SET ALSO VARIES FROM
# 27 FILE TO FILE. ALL PERMANENT FILES ARE CREATED BY TRANSFERRING
# 28 INFORMATION FROM DATA CARDS TO DISK STORAGE. THE UNITS FILE IS
# 29 ESTABLISHED BY DIRECT USER INPUT, WHILE THE NODE+VECTOR FILE
# 30 IN MEMORY AND THE DIRECTORIES IN MEMORY ARE CREATED BY THE PRO-
# 31 GRAM, BUT ONLY AFTER THE USER HAS INVOKED PARTICULAR ROUTINES.
#####
# 32
# 33
# 34
# 35
# 36
# 37
# 38
# 39
# 40
# 41
# 42
# 43
# 44
# 45
# 46
# 47
#####
```



GROUND MOVEMENT CALCULATIONS

```
#####
# 1 THERE ARE TWO BASIC TYPES OF GROUND MOVEMENT CALCU- 49
# 2 LATIONS WE WILL MAKE IN THIS PACKAGE. ONE CONSIDERS THE 50
# 3 ROAD NETWORK FOR THE AREA OF INTEREST AS REPRESENTED IN THE 51
# 4 NODE+VECTOR DATA BASE, WHILE THE OTHER USES CROSS-COUNTRY 52
# 5 MOVEMENT INFORMATION AS SUPPLIED BY THE ARMY ON ITS COM AND 53
# 6 CONCEALMENT MAPS FOR THE AREA OF INTEREST. 54
# 7 THE MAJOR DIFFERENCES BETWEEN THESE TWO APPROACHES 55
# 8 ARE IN THEIR SCOPE OF APPLICATION, AND THE DATA BASES WHICH 56
# 9 SUPPLY THE INFORMATION. USING THE ROAD NETWORK APPROACH, ONE 57
# 10 MAY CONSIDER UP TO TEN 1:50000 MAPS FOR ANY COMPUTATION, WHILE 58
# 11 THE CROSS-COUNTRY MOVEMENT MUST BE LIMITED TO AT MOST ONE 59
# 12 1:50000 MAP. THIS IS DUE TO THE MANNER IN WHICH THE TWO TYPES 60
# 13 ARE MODELED. 61
# 14 BECAUSE OF THE DIFFERENCE IN INFORMATION BEING HELD IN 62
# 15 THE DATA BASE REGARDING THE TWO TYPES OF MOVEMENT, THIS CALLS 63
# 16 FOR TWO VERY DIFFERENT DATA BASE APPROACHES. THESE WILL BE 64
# 17 EXPLAINED LATER ON IN THIS DOCUMENT. 65
# 18 ON THE OTHER HAND, THE ALGORITHM USED TO CALCULATE THE 66
# 19 PATHS FOR EACH TYPE OF MOVEMENT IS VIRTUALLY IDENTICAL. THE 67
# 20 ONLY DIFFERENCE LIES IN THE INTERPRETATION OF THE DATA BASES; 68
# 21 THE ACTUAL MECHANICS OF THE PATH CALCULATIONS ARE THE SAME. 69
# 22 THE ALGORITHM FINDS THE "BEST" PATH FOR A UNIT TO TRAVEL. 70
# 23 "BEST" IS A USER-DEFINED PARAMETER MADE UP OF TWO COMPONENTS, 71
# 24 RISK AND TIME. RISK IS A VALUE ARRIVED AT BY CONSIDERING 72
# 25 ROAD CONDITIONS, TERRAIN FACTORS, WHETHER OR NOT A BRIDGE IS 73
# 26 PRESENT, AND WHETHER OR NOT A CITY IS PRESENT. TIME IS THE 74
# 27 AMOUNT OF TIME IT TAKES A PARTICULAR UNIT TO TRAVEL FROM ONE 75
# 28 NODE TO ANOTHER. 76
# 29 OTHER TYPES OF PATH CALCULATIONS MAY BE PERFORMED AS 77
# 30 WELL. THE USER MAY FIND THE SECOND BEST PATH TO TRY TO ATTACK 78
# 31 THE SENSOR TASKING PROBLEM, HE MAY FIND THE BEST NODE AT WHICH 79
# 32 TO PERFORM INTERDICTION FOR DISRUPTION PURPOSES, OR HE MAY FIND 80
# 33 THE BEST LONGEST PATH (A METHOD OF CALCULATING HOW QUICKLY A FORCE 81
# 34 MAY BUILD UP IN A PARTICULAR AREA. 82
#####
```

TRN, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
GENERAL DESCRIPTION

PAGE 6

USER FORMAT

```
#####  
# 1 THE FORMAT FOR USER INTERACTION WILL BE OF A SIMPLE NATURE. 84  
# 2 WHEN STARTING, A BASIC MENU OF AVAILABLE FUNCTIONS WILL BE LISTED ON 85  
# 3 A DISPLAY SCREEN. THE USER WILL DECIDE WHICH ONE HE WISHES TO PURSUE, 86  
# 4 AND ENTER ITS CODE AT A TERMINAL. THIS MAY CAUSE DIRECT ACTION, OR 87  
# 5 CAUSE ANOTHER MENU TO BE DISPLAYED. THROUGHOUT THE OPERATION, WHETHER 88  
# 6 OR NOT A MENU APPEARS, THE USER WILL BE PROMPTED WITH INSTRUCTIONS 89  
# 7 REGARDING HOW TO PROCEED. IN THIS WAY, THE USER HAS 'HANDS-ON' CONTROL 90  
# 8 OF ANY AND ALL ACTIVITIES INVOLVED IN THE GROUND FORCE MOVEMENT COMPU- 91  
# 9 TATIONS. 92  
#####
```



# SYSTEM ARCHITECTURE

```

# 1 THIS SOFTWARE HAS BEEN DESIGNED TO OPERATE ON A PDP 11/45 COMPUTER
# 2 HAVING 96K WORDS OF MEMORY, WITH ANY SINGLE TASK ABLE TO ACCESS AT MOST
# 3 32K WORDS. THE OPERATING SYSTEM, RSX-11M, SUPPORTS A USER DEFINED
# 4 MULTITASKING CAPABILITY, WITH THE TOTAL AMOUNT OF MEMORY USAGE TO NOT
# 5 EXCEED 64K WORDS.
# 6 FOUR DIFFERENT PERIPHERALS ARE NEEDED TO EFFECTIVELY USE
# 7 THIS SOFTWARE PACKAGE. ONE IS A TERMINAL FOR COMMUNICATION WITH THE
# 8 PROGRAM. THIS MAY BE A DECURITER, A TEKTRONIX CRT, OR A VT05 TERMINAL.
# 9 ALL PERMANENT DATA BASES ARE STORED IN AN RPO4 MOVING HEAD DISK UNIT, AS
# 10 WELL AS ALL OF THE SOURCE AND OBJECT CODE, AND TASK IMAGE. A GOULD 4800
# 11 HIGH-SPEED LINE PRINTER IS USED FOR PATH STATISTICS OUTPUT, AND A COMTEL
# 12 8500 IMAGE PROCESSING SYSTEM IS USED FOR THE VISUAL DISPLAYS.
# 13 DUE TO MEMORY CONSIDERATIONS, SUCH AS THE LARGE AMOUNT OF SPACE
# 14 NEEDED FOR DATA BASES, IT HAS BEEN DECIDED TO USE MULTIPLE TASKING TO
# 15 IMPLEMENT THIS PROGRAM. ONE TASK PERFORMS ALL OF THE I/O FUNCTIONS, THE
# 16 OBTAINING OF USER INPUT AND THE PRINTING OF MESSAGES, WHILE ANOTHER DOES
# 17 ALL OF THE COMPUTATION. RSX-11M IS STRUCTURED SO THAT THIS MAY BE DONE
# 18 WITH A MINIMUM OF DELAY TIME USED TO SWITCH FROM ONE TASK TO THE OTHER.
# 19 MESSAGES MAY BE SENT BETWEEN THE TWO, AND THIS IS HOW OUTPUT DATA IS
# 20 TRANSFERRED. THERE IS ALSO A SEPARATE GRAPHICS PACKAGE WHICH IS ITSELF
# 21 ANOTHER TASK, SO THERE ARE THREE TASKS RUNNING CONCURRENTLY IN THIS SYS-
# 22 TEM.
# 23 THE TOTAL MEMORY OF THESE THREE TASKS MUST NOT EXCEED 64K WORDS,
# 24 OR SEVERE DEGRADATION OF RESPONSE TIME IS INCURRED. THIS NECESSITATES
# 25 THE USE OF OVERLAYS IN ORDER TO SAVE MEMORY SPACE.
# 26 WE HAVE ARRIVED AT THESE ESTIMATES FOR STORAGE REQUIREMENTS:
# 27
# 28
# 29
# 30
# 31
# 32
# 33
# 34
# 35
# 36

```

DESCRIPTION	CONTENTS	MEMORY USED
IN-CORE NODE-ARRAY	400 NODES AT 32 WORDS EACH (ABOUT 10 MAPS)	12,800 WORDS
SOLUTION-VECTORS	40 ROUTES WITH 25 NODES PER ROUTE.	5,000 WORDS

TRU, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
GENERAL DESCRIPTION

PAGE 7.001

# 37	OTHER	DIRECTORIES, COUNTERS, ETC.	5,000 WORDS	#	130	
# 38	.			#	131	
# 39	TOTAL		22,800 WORDS	#	132	
# 40	.			#	133	
# 41	.			#	134	
# 42	.			#	135	
# 43		<p>THIS APPARENTLY LEAVING US WITH 10,000 WORDS FOR ACTUAL EXECUTABLE CODE PER OVERLAY. HOWEVER, THE GRAPHICS PACKAGE USES 21,300 WORDS, AND THE I/O ROUTINES USE APPROXIMATELY 17,000 WORDS, SO ONLY ABOUT 4,000 WORDS ARE TRULY AVAILABLE FOR CODING IN THE MAIN ROUTINE.</p>			#	136
# 44				#	137	
# 45				#	138	
# 46				#	139	
# 47	TIMINGS:	TO BE DETERMINED		#	140	
#				#		



TRM, INC.  
21 FEB '8

LOCATION AND MOVEMENT ANALYSIS SYSTEM

PAGE 8

\*\*\*\*\*  
\*  
\* DATA BASE DESCRIPTIONS \*  
\*  
\*\*\*\*\*

NODES AND LINKS

```
#####
# 1 CONSISTING OF ONLY NODE+VECTORS. THIS DATA BASE IS THE LARGEST, AND
# 2 PROBABLY THE MOST IMPORTANT, OF ALL THE DATA BASES. THE INFORMATION
# 3 CONTAINED WITHIN ENABLES THE PATH ALGORITHM TO ACCURATELY CALCULATE THE
# 4 BEST PATH FOR A GIVEN UNIT TO TRAVEL.
# 5 A NODE+VECTOR IS A 32-WORD STORAGE AREA WHICH CONTAINS DATA PER-
# 6 TAINING TO A NODE AND THE LINKS WHICH CONNECT IT TO OTHER NODES. A NODE
# 7 IS A POINT OF INTEREST, USUALLY, BUT NOT NECESSARILY, BEING AT THE JUNC-
# 8 TION OF TWO OR MORE ROADS. THE LINK BETWEEN TWO NODES IS MOST OFTEN A
# 9 REPRESENTATION OF THE ROADWAY JOINING THE TWO, BUT IT CAN ALSO BE A REP-
# 10 RESENTATION OF SIMPLY THE GENERAL PATH (CROSS-COUNTRY OR OTHERWISE)
# 11 WHICH CONNECTS THEM. TWO NODES ARE SAID TO BE ADJACENT TO EACH OTHER
# 12 IF THERE EXISTS A SINGLE LINK BETWEEN THEM. A PATH IS THEN A SERIES
# 13 OF LINKS SUCH THAT THE LINKS, WHEN TAKEN AS A WHOLE, ARE CONNECTED. THAT
# 14 IS, STARTING AT NODE 1, A LINK CONNECTS WITH NODE 2 ADJACENT TO NODE 1,
# 15 WHICH CONNECTS WITH NODE 3 ADJACENT TO NODE 2, ETC., UNTIL THE DESTI-
# 16 NATION NODE IS REACHED. ALL NODE+VECTOR DATA HAS BEEN OBTAINED FROM
# 17 STANDARD L SERIES 1:50000 MAPS.
# 18 A GROUND FORCE UNIT'S MOVEMENT THROUGH A ROAD NETWORK IS CALCULATED
# 19 BY MODELING ITS TRAVEL FROM ONE NODE TO ANOTHER ALONG A LINK. A LINK
# 20 IS A CHARACTERIZATION OF A PATH, AND AS SUCH, PASSES ALONG ASSOCIATED
# 21 PIECES OF INFORMATION, SUCH AS: DISTANCE, NUMBER OF LANES, ROAD TYPE,
# 22 OFF-ROAD TRAFFICABILITY, AND TERRAIN, BRIDGE, AND CITY CODES.
# 23 DISTANCE REFERS TO THE ACTUAL DISTANCE IN KILOMETERS ONE WOULD
# 24 HAVE TO TRAVEL TO MOVE FROM ONE NODE TO ANOTHER ALONG THE TRUE CONNEC-
# 25 TION. NUMBER OF LANES MEANS THE NUMBER OF SINGLE-FILE ROAD LANES
# 26 WHICH ARE EQUIVALENT TO THE LINKING ROAD. ROAD TYPE IS A CODE REFLECT-
# 27 ING ANY OF FOUR POSSIBILITIES: AUTOBARN OR AUTOSTAGE, MAIN ROAD, SEC-
# 28 ONARY ROAD, OR FAIR WEATHER ROAD ONLY. OFF-ROAD TRAFFICABILITY IS AND-
# 29 THER CODE, THIS TIME INDICATING OPEN TRAFFICABLE AREAS NEAR THE ROAD
# 30 WHICH WILL ALLOW REST OR ALTERNATE ROUTES. TERRAIN MEASURES LAND FEAT-
# 31 URES AND INTERVISIBILITY. THE CHOICES ARE FLAT, HILLY, MOUNTAINOUS, OR
# 32 ANY OF THESE THREE PLUS HIGH TERRAIN IMMEDIATELY TO THE WEST. THE
# 33 BRIDGE CODE INDICATES THE PRESENCE, OR LACK OF, A BRIDGE, AND IF IT EX-
# 34 ISTS, ITS SIZE, EITHER SMALL, MEDIUM, OR LARGE. FINALLY, CITY CODE
# 35 FUNCTIONS IN A MANNER IDENTICAL TO THE BRIDGE CODE. RISK IS TAKEN TO
# 36 BE A LINEAR COMBINATION OF OFF-ROAD TRAFFICABILITY, TERRAIN, BRIDGE,
#####
# 143
# 144
# 145
# 146
# 147
# 148
# 149
# 150
# 151
# 152
# 153
# 154
# 155
# 156
# 157
# 158
# 159
# 160
# 161
# 162
# 163
# 164
# 165
# 166
# 167
# 168
# 169
# 170
# 171
# 172
# 173
# 174
# 175
# 176
# 177
# 178
```



LOCATION AND MOVEMENT ANALYSIS SYSTEM  
DATA BASE DESCRIPTIONS

AND CITY CODES.

THE CROSS-COUNTRY AND CONCEALMENT PATH ALGORITHM DOES NOT USE THE NODES AND LINKS DATA BASE.

THE ELEMENTS WHICH CHARACTERIZE A NODE ARE SIGNIFICANTLY DIFFERENT THAN THOSE CHARACTERIZING A LINK. ALL NODAL INFORMATION IS LOCATION ORIENTED, THAT IS, IT DESCRIBES EXACTLY WHERE THE NODE IS ON THE MAP. THIS INFORMATION INCLUDES LATITUDE, LONGITUDE, THE MAP NUMBER FOR THE PARTICULAR MAP ON WHICH THE NODE RESIDES, AND THE NODE NUMBER (AN ARBITRARY FIGURE ESTABLISHED DURING THE MANUAL CREATION OF THE DATA BASE.)

A NODE+VECTOR THEN, IS COMPOSED OF THE ABOVEMENTIONED NODAL AND LINK INFORMATION (A MAXIMUM OF FOUR ADJACENT NODES MAY EXIST), PLUS SEVERAL OTHER VALUES, WHOSE SIGNIFICANCE WILL BECOME CLEAR LATER ON IN THIS DOCUMENT (AMONG THESE ARE NORTH MEASURE, PREDECESSOR NODE, CUMULATIVE TIME, CUMULATIVE NORTH MEASURE, TIME MEASURE, SOLUTION VECTOR POINTER, LANE USED, RISK MEASURE, AND PARK TIME. THESE ARE ALL USED BY THE PATH ALGORITHM.)

NODE+VECTORS MAKE UP A PERMANENT DATA BASE WHICH RESIDES ON DISK. ONE MAY THINK OF THIS STORAGE AS BEING DIVIDED INTO BINS; SUCH THAT ONE BIN CONTAINS THE NODE+VECTORS FOR ONE MAP. A DIRECTORY IS ALSO STORED ON DISK, AND IT CONTAINS IDENTIFICATION FOR EACH BIN, THAT IS, THE APPROPRIATE MAP NUMBER. ACCESS OF ANY MAP'S NODE+VECTORS IS QUITE SIMPLE; ONE FINDS THE MAP NUMBER IN THE DIRECTORY. THE DIRECTORY ENTRY POINTS TO THE BIN ON DISK WHICH CONTAINS THE NODE+VECTORS, WHICH MAY THEN BE READ INTO MAIN MEMORY.

ONCE IN MAIN MEMORY, NODE+VECTORS ARE ALTERABLE BY THE USER. FUNCTIONS EXIST WHICH ALLOW THE USER TO CHANGE CERTAIN VALUES: SUCH AS LINK DISTANCE, ROAD TYPE, OR NUMBER OF LANES. ANY CHANGES MADE ARE NOT PERMANENT; SINCE THE "CORRECTED" VERSION OF THE DATA BASE IS NEVER COPIED INTO THE PERMANENT DISK FILE.

WHILE IN MEMORY, THE NODE+VECTORS ARE THOUGHT OF AS OCCUPYING A NODE+VECTOR ARRAY. MAIN MEMORY IS LINEARLY ORGANIZED, SO IT IS NATURAL TO THINK OF NODE+VECTORS COMPRISING A LIST INDEXABLE BY INTEGERS. THUS, IF TWO MAPS' NODE+VECTORS ARE IN MEMORY, WITH THE FIRST MAP HAVING 30 NODE+VECTORS, THE FIRST NODE+VECTOR OF THE SECOND MAP IS THE 31 ST NODE+VECTOR IN THE ARRAY, ETC.

#####





TRU, INC. LOCATION AND MOVEMENT ANALYSIS SYSTEM  
 21 FEB 78 DATA BASE DESCRIPTIONS

PAGE 10.001

# 37	29	# OF LINES	;	ROAD TYPE	#	251
# 38	30	OFF-ROAD TRAFFICABILITY	;	TERRAIN	#	252
# 39	31	BRIDGE	;	CITY	#	253
# 40	32	RISK MEASURE	;	PARK TIME	#	254
#			;		#	

TRW, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
DATA BASE DESCRIPTIONS

PAGE 11

NODE VECTOR DESCRIPTION

#	WORD #	DESCRIPTION AND USE	RANGE
# 1	1	NUMBER OF 1:50000 MAP FROM WHICH DATA IS TAKEN.	1-32767
# 2	2,1	NUMBER OF NODE WITHIN THE MAP. THESE NUMBERS ARE SELECTED WHEN THE NODES ARE MANUALLY DEFINED.	1-80
# 3	2,2	FLAG TO INDICATE WHETHER OR NOT THIS NODE HAS BEEN LABELED BY THE PATH ALGORITHM.	0-1
# 4	3	WORD OF THE LINK BETWEEN THIS NODE AND ITS PREDECESSOR, AS DETERMINED BY THE PATH ALGORITHM.	0-100
# 5	4	NODE+VECTOR INDEX OF THE PREDECESSOR NODE+VECTOR, SELECTED BY THE OPTIMAL MOVE ALGORITHM.	1-400
# 6	5	THE CUMULATIVE TIME, IN MINUTES, OF A MOVEMENT UNIT AT THIS NODE POSITION AS DETERMINED BY THE OPTIMAL MOVEMENT ALGORITHM. THERE IS A MAXIMUM TRAVEL TIME OF TEN DAYS (=14400 MINUTES) PLUS 23 HOURS, 59 MINUTES (=1439 MINUTES).	0-15839
# 7	6	TOTAL WORD MEASURE OF THE PATH FROM INITIAL NODE UP TO AND INCLUDING THIS LINK'S WORD MEASURE (WORD 3).	0-32767
# 8	7	THE ACTUAL TIME FOR A UNIT TO PROGRESS FROM PREDECESSOR TO THIS NODE.	0-15839
# 9	8	INDEX TO ENTRY IN THE SOLUTION+VECTOR ARRAY. THIS POINTER ENABLES THE NODE+VECTOR TO BE ASSOCIATED WITH THE OPTIMAL ROUTE DATA. ONLY NODES THAT ARE PART OF AN OPTIMAL SOLUTION WILL HAVE THIS ENTRY. IF THE NODE IS PART OF SEVERAL OPTIMAL ROUTES, THIS WORD WILL POINT TO THE	0-5000



TRW, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
DATA BASE DESCRIPTIONS

PAGE 11.001

# 37	MOST RECENTLY CALCULATED SOLUTION+VECTOR.		# 292
# 38			# 293
# 39	9 NODE LATITUDE IN KILOMETERS FROM EQUATOR.	0-32767	# 294
# 40	THIS DATA IS ONLY USED FOR DISPLAY AND IS THUS		# 295
# 41	+ OR - 1 KM RESOLUTION. ROUTE ALGORITHMS USE LINK		# 296
# 42	DISTANCES, NOT THESE LATITUDE, LONGITUDE COOR-		# 297
# 43	DINATES.		# 298
# 44			# 299
# 45	10 NODE LONGITUDE IN KILOMETERS FROM GREENWICH.	0-32767	# 300
# 46	THIS DATA IS ONLY USED FOR DISPLAY AND IS THUS		# 301
# 47	+ OR - 1 KM RESOLUTION.		# 302
# 48			# 303
# 49	11.1 NUMBER OF ADJACENT NODES, MAXIMUM IS FOUR,	1-4	# 304
# 50	MINIMUM IS ONE.		# 305
# 51			# 306
# 52	11.2 LANE NUMBER USED DURING PATH CALCULATION. THIS	1-4	# 307
# 53	IS NECESSARY IN ORDER TO EFFECTIVELY USE MULTIPLE		# 308
# 54	LANED ROADS.		# 309
# 55			# 310
# 56	12 THE MAP+NUMBER OF THE FIRST ADJACENT NODE.	1-32767	# 311
# 57	THIS ENTRY WILL BE RESET AFTER DATA IS READ		# 312
# 58	FROM DISK TO CURE TO BECOME THE NODE+VECTOR		# 313
# 59	ARRAY INDEX OF THE FIRST ADJACENT NODE.		# 314
# 60			# 315
# 61	13.1 NUMBER OF NODE WITHIN THE MAP OF THE FIRST	1-60	# 316
# 62	ADJACENT NODE.		# 317
# 63			# 318
# 64	13.2 LINK DISTANCE IN TENTHS OF A KM BETWEEN THE	0-255	# 319
# 65	NODE AND THE FIRST ADJACENT NODE.		# 320
# 66			# 321
# 67	14.1 NUMBER OF SINGLE FILE ROAD LANES BETWEEN	1-4	# 322
# 68	THE NODE AND THE FIRST ADJACENT NODE.		# 323
# 69			# 324
# 70	14.2 ROAD TYPE CODE. 1-AUTOBRAIN OR AUTOSTRADE,	1-4	# 325
# 71	2-MAIN ROAD, 3-SECONDARY ROAD OR ROAD, 4-		# 326
# 72	FAIR WEATHER ROAD ONLY.		# 327
# 73			# 328
# 74	15.1 GROUND COVER CODE TO INDICATE OPEN TRAFFIC-	1-5	# 329

# 75	ABLE AREAS NEAR ROAD THAT ALLOWS REST OR ALTER-	#	330
# 76	NATE PATHS.	#	331
# 77	1=URBAN, 2=CULTIVATED, 3=WOODED, 4=SWAMP,	#	332
# 78	5=ROAD NET ADJACENT TO MAIN LINK.	#	333
# 79		#	334
# 80	15,2 TERRAIN. CODE TO INDICATE TERRAIN AND COARSE	#	335
# 81	INTERVISIBILITY MEASURE. 1=FLAT, 2=HILLY,	#	336
# 82	3=MOUNTAINOUS, 10=HIGH TERRAIN IMMEDIATELY TO THE	#	337
# 83	WEST, AND ANY OF THE FIRST THREE PLUS 10.	#	338
# 84		#	339
# 85	16,1 BRIDGE. 0=NONE, 1=SMALL, 2=MEDIUM, 3=LARGE.	#	340
# 86		#	341
# 87	16,2 CITY. 0=NONE, 1=SMALL, 2=MEDIUM, 3=LARGE	#	342
# 88		#	343
# 89	17-21 IDENTICAL TO WORDS 12-16 BUT APPLICABLE TO THE	#	344
# 90	SECOND ADJACENT NODE, IF PRESENT.	#	345
# 91		#	346
# 92	22-26 IDENTICAL TO WORDS 12-16 BUT APPLICABLE TO THE	#	347
# 93	THIRD ADJACENT NODE, IF PRESENT.	#	348
# 94		#	349
# 95	27-31 IDENTICAL TO WORDS 12-16 BUT APPLICABLE TO THE	#	350
# 96	FOURTH ADJACENT NODE, IF PRESENT.	#	351
# 97		#	352
# 98	32,1 RISK MEASURE FOR THE LINK BETWEEN THIS NODE AND	#	353
# 99	ITS PREDECESSOR. RISK IS TAKEN TO BE THE SUM OF	#	354
# 00	GROUND COVER, TERRAIN, BRIDGE, AND CITY CODES.	#	355
# 01		#	356
# 02	32,2 USED BY THE PATH ALGORITHM TO REMEMBER HOW MANY	#	357
# 03	MINUTES A UNIT PARKED AT THIS NODE.	#	358
#		#	
#	#####	#	
#	TOO MANY LINES IN SEGMENT	#	



## TERRAIN

THIS DATA BASE SUPPORTS THE TERRAIN MODEL FOR PATH CALCULATIONS. THERE ARE ACTUALLY THREE DATA BASES INVOLVED, CROSS-COUNTRY MOVEMENT, SUMMER CONCEALMENT, AND WINTER CONCEALMENT.

EACH DATA BASE SUPPLIES DATA FOR A SPECIFIC TYPE OF PATH CALCULATION. CROSS-COUNTRY YIELDS DATA FOR FASTEST MOVEMENT, AND THE TWO CONCEALMENT DATA BASES GIVE INFORMATION ABOUT THE RISK OF A PATH.

THESE DATA BASES WERE CREATED BY PLACING A 93 X 88 SQUARE GRID OVER THE DATA MAPS SUPPLIED BY THE ARMY. EACH SQUARE WAS THEN ASSIGNED A VALUE CORRESPONDING TO THE AREA WHICH THE SQUARE COVERED. THESE VALUES WERE TRANSFERRED TO COMPUTER CARDS WHICH WERE THEN USED TO CREATE THREE DATA FILES ON DISK, 'CCM.DAT', 'SCON.DAT', AND 'WCON.DAT'.

BECAUSE OF THE REGULARITY OF THE SQUARES, EACH SQUARE MAY BE REGARDED AS A NODE WITH EIGHT NEIGHBORS (UNLESS IT LIES ON AN EDGE). THUS, EACH MAP IS REPRESENTED BY 8184 NODES. FOR ANY NODE, NO MORE INFORMATION IS NEEDED, SINCE ALL LINK ATTRIBUTES ARE KNOWN.

THERE IS, HOWEVER, PATH INFORMATION WHICH MUST BE STORED DURING A CALCULATION, BUT THERE SIMPLY ISN'T ENOUGH MEMORY TO ALLOCATE 5 WORDS FOR EACH OF THE 8184 NODES. THUS, WHEN USING THE TERRAIN MODEL FOR PATH CALCULATIONS, THERE IS AN ADDITIONAL DATA BASE CALLED THE 'WORKING LIST'. AS A NEW NODE IS INCLUDED IN THE CALCULATION, A NEW ENTRY TO THE WORKING LIST IS MADE. SINCE NOT ALL NODES WILL BE INCLUDED, THE WORKING LIST'S SIZE IS CONSIDERABLY SMALLER THAN 8184 \* 5 WORDS, AND FITS IN MAIN MEMORY WITH EASE.

ANOTHER SPACE-SAVING DEVICE IS EMPLOYED. THE CODE FOR ANY SQUARE IS NEVER GREATER THAN 15, SO DATA FOR FOUR NODES MAY BE PLACED IN ONE WORD OF MEMORY, THUS 8184 CODES USE ONLY 2046 WORDS.

TRW, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
DATA BASE DESCRIPTIONS

PAGE 13

CROSS-COUNTRY AND CONCEALMENT CODE DESCRIPTIONS

```
#####
# 1 THESE CODES COME DIRECTLY FROM THE MAPS PREPARED BY THE TERRAIN
# 2 ANALYSIS CENTER, US ARMY ENGINEER TOPOGRAPHIC LABORATORIES, FORT
# 3 BELVOIR, VIRGINIA, IN SEPTEMBER, 1977.
# 4
# 5
# 6
# 7
# 8
# 9
# 10
# 11
# 12
# 13
# 14
# 15
# 16
# 17
# 18
# 19
# 20
# 21
# 22
# 23
# 24
# 25
# 26
#####

CROSS-COUNTRY MOVEMENT

CODE      ESTIMATED MAX. SPEED OF XM-1  MANEUVERABILITY
          TANK (MPH)                  DESCRIPTOR
1         >= 25                      EXCELLENT
2         20 - 25                    VERY GOOD
3         10 - 20                    GOOD
4         5 - 10                    FAIR
5         < 5                       POOR
6         PASSAGE BLOCKED            NC GO
7         BUILT-UP AREA (TRAVEL OF 7 ROADS NOT EVALUATED)

CONCEALMENT

CODE      PERCENT CHANCE OF BEING OBSERVED FROM THE AIR
8         0 - 25 (BEST CONCEALMENT)
9         25 - 50
10        50 - 75
11        75 - 100 (POOREST CONCEALMENT)

#####
# 387
# 388
# 389
# 390
# 391
# 392
# 393
# 394
# 395
# 396
# 397
# 398
# 399
# 400
# 401
# 402
# 403
# 404
# 405
# 406
# 407
# 408
# 409
# 410
# 411
# 412
#####
```



## SLIMS

```

# # # # #
# 1 THIS IS A TEMPORARY DATA BASE, THAT IS, A DATA BASE WHICH MAY
# 2 CHANGE FROM USER TO USER, AND, IN FACT, FROM ONE PATH CALCULATION TO
# 3 ANOTHER, IF DESIRED. BECAUSE OF ITS TEMPORARY NATURE, IT EXISTS ONLY
# 4 IN MAIN MEMORY; THERE IS NO UNITS DATA BASE ON DISK.
# 5 THE UNIT+VECTOR IS THE SOLE CONSTITUENT WITHIN THIS DATA BASE. IT
# 6 CONTAINS INFORMATION PERTAINING TO A UNIT, SUCH AS ITS NAME, STARTING
# 7 LOCATION, DESTINATION LOCATION, PRESENT LOCATION, PRIORITY, START TIME,
# 8 AND TYPE CODE. EACH LOCATION IS ENTERED BY THE USER AS A MAP+NUMBER,
# 9 NODE+NUMBER PAIR, WHERE THE MAP+NUMBER IS THE NUMBER OF THE 1:50000 MAP
# 10 ON WHICH THE NODE LIES, AND THE NODE+NUMBER IS THAT NUMBER ARBITRARILY
# 11 ASSIGNED WHEN THE NODE+VECTOR DATA BASE WAS CONSTRUCTED. HOWEVER, FOR
# 12 PURPOSES OF STORAGE, THESE LOCATIONS ARE REPRESENTED INTERNALLY AS IN-
# 13 DICES INTO THE NODE+VECTOR ARRAY. PRIORITY IS A NUMBER REPRESENTING
# 14 A UNIT'S IMPORTANCE WITH RESPECT TO OTHER UNITS. START TIME IS A VALUE
# 15 ENTERED BY A PATH ALGORITHM, NOT THE USER, AND REPRESENTS MINUTES OF
# 16 TIME. TYPE CODE IS A NUMBER WHICH STANDS FOR THE PARTICULAR TYPE OF
# 17 UNIT BEING CONSIDERED, SUCH AS A RIFLE BATTALION, OR A MOTORIZED DIVI-
# 18 SION.
# 19
# 20 THIS DATA BASE MAY BE RECREATED EACH TIME A USER WISHES TO PER-
# 21 FORM A PATH CALCULATION, AND MAY CONTAIN ANYWHERE FROM 1 TO 60 SEPARATE
# 22 UNITS. IF NO UNITS ARE ESTABLISHED BY THE USER, THE PATH ALGORITHMS
# 23 WILL NOT FUNCTION, AND AN ERROR MESSAGE PRINTED.
# # # # #

```

UNIT VECTOR FORMAT

WORD #	BYTE 1	BYTE 2	
1	FIRST LETTER	SECOND LETTER	437
2	THIRD LETTER	FOURTH LETTER	438
3	FIFTH LETTER	SIXTH LETTER	439
4	SEVENTH LETTER	EIGHTH LETTER	440
5	STARTING LOCATION		441
6	DESTINATION LOCATION		442
7	PRESENT LOCATION		443
8	PRIORITY		444
9	START TIME		445
10	SPARE		446
11			447



TRM, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
DATA BASE DESCRIPTIONS

PAGE 15

UNIT+VECTOR DESCRIPTION

```
#####  
# 1 WORD # DESCRIPTION AND USE RANGE  
# 2 1-4 THIS IS THE UNIT'S NAME IN CHAR- ANY ALLUABLE  
# 3 ACTER FORMAT. THERE MAY BE AT CHARACTER  
# 4 MOST EIGHT CHARACTERS.  
# 5  
# 6 AN INDEX INTO THE NODE+VECTOR 1-400  
# 7 ARRAY. THUS, THIS WORD SAYS THAT  
# 8 IF THE INDEX EQUALS I, THE PROPER  
# 9 NODE IS THE I TH NODE IN THE ARRAY.  
# 10  
# 11 AN INDEX INTO THE NODE+VECTOR 1-400  
# 12 ARRAY, INDICATING THE DESTINATION  
# 13 OF THIS UNIT.  
# 14  
# 15 AN INDEX INTO THE NODE+VECTOR 1-400  
# 16 ARRAY, INDICATING WHERE THE UNIT  
# 17 IS AT THIS MOMENT IN THE CALCULATION.  
# 18  
# 19 PRIORITY OF THIS UNIT. 1 IS THE HIGH- 0-127  
# 20 EST, 127 IS THE LOWEST. 0 INDICATES  
# 21 THAT THE PRIORITY HAS NOT BEEN ES-  
# 22 TABLISHED.  
# 23  
# 24 CODE INDICATING GENERAL TYPE 0-255  
# 25 OF UNIT.  
# 26  
# 27 TIME AT WHICH UNIT IS KNOWN TO BE AT 0-15839  
# 28 EITHER ITS STARTING OR FINISHING NODE.  
# 29 THIS DEPENDS UPON MOVEMENT TYPE.  
# 30  
# 31 SPACE  
#####  
# 449  
# 450  
# 451  
# 452  
# 453  
# 454  
# 455  
# 456  
# 457  
# 458  
# 459  
# 460  
# 461  
# 462  
# 463  
# 464  
# 465  
# 466  
# 467  
# 468  
# 469  
# 470  
# 471  
# 472  
# 473  
# 474  
# 475  
# 476  
# 477  
# 478  
# 479
```

## SOLUTIONS

```
#####  
# 1 ANOTHER TEMPORARY DATA BASE, THIS IS CREATED BY THE PATH ALGORITHM  
# 2 AS IT RECONSTRUCTS A ROUTE OF TRAVEL. EACH SOLUTION+VECTOR SO CREATED  
# 3 CONTAINS INFORMATION ABOUT A PARTICULAR NODE; ITS NODE+VECTOR INDEX,  
# 4 TIME IN, TIME OUT, LANE NUMBER, AND PARK TIME, AND FURTHER DIAGNOSTIC  
# 5 DATA: A POINTER TO ANOTHER SOLUTION+VECTOR WHICH CONSIDERS THE SAME  
# 6 NODE+VECTOR (IF NONE, THE POINTER IS NULL), AND WHETHER OR NOT THIS SOL-  
# 7 UTION+VECTOR SHOULD BE USED BY THE PATH DECONFLICTING LOGIC.  
# 8 A NODE+VECTOR INDEX IS A POINTER TO A NODE+VECTOR IN MAIN MEMORY.  
# 9 TIME IN AND TIME OUT INDICATE A TIME PERIOD DURING WHICH THE NODE+VEC-  
# 10 TOR IS BUSY. LANE NUMBER KEEPS TRACK OF WHICH LANE THIS UNIT USED TO  
# 11 TRAVEL THROUGH THIS NODE. THIS ENABLES MULTIPLE LANE ROADS TO BE USED  
# 12 BY MORE THAN ONE UNIT AT A TIME. PARK TIME SHOWS HOW MUCH TIME (IN MIN-  
# 13 UTES) THE UNIT HAD TO WAIT AT THIS NODE'S PREDECESSOR BEFORE TRAVELING  
# 14 TO THIS NODE.  
# 15 THE SOLUTION+VECTOR POINTER ALLOWS THE DECONFLICTING LOGIC TO OPER-  
# 16 ATE. BY FOLLOWING THE POINTERS, IT IS POSSIBLE TO FIND ALL INTERVALS  
# 17 DURING WHICH A PARTICULAR NODE IS BUSY. IN THIS WAY, DECONFLICTING THE  
# 18 PROCESS OF MAKING SURE THAT NO TWO UNITS USE A NODE DURING OVERLAPPING  
# 19 TIME INTERVALS, MAY BE ACCOMPLISHED. THE USER HAS THE OPTION TO INCLUDE  
# 20 OR EXCLUDE THE DECONFLICTING LOGIC. DATA WITHIN THE SOLUTION+VECTOR  
# 21 REMEMBERS THE USER'S CHOICE.  
# 22 THE ROUTE DESCRIBED BY A PATH IS A SERIES OF SOLUTION+VECTORS  
# 23 HAVING A ONE-TO-ONE CORRESPONDENCE WITH THE NODES OF THE PATH. IF A  
# 24 PATH'S NODES WERE NUMBERED 1-N, ITS SOLUTION+VECTORS WOULD BE NUM-  
# 25 BERED M+1-M+N, ASSUMING M SOLUTION+VECTORS ALREADY EXISTED AT THE TIME  
# 26 THE PATH WAS CALCULATED. NODE 1 WOULD CORRESPOND TO SOLUTION+VECTOR  
# 27 M+1, NODE 2 WOULD CORRESPOND TO SOLUTION+VECTOR M+2, ETC.  
#####  
#
```

481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507





TRM, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
DATA BASE DESCRIPTIONS

PAGE 19

SOLUTION+VECTOR DESCRIPTION

```
#####  
# 1 WORD # DESCRIPTION AND USE RANGE  
#####  
# 2 INDEX NUMBER OF THE ASSOCIATED NODE+VECTOR.  
# 3 SEVERAL ROUTES FOR DIFFERENT UNITS MAY POINT  
# 4 TO THE SAME NODE+VECTOR.  
# 5  
# 6  
# 7 THIS IS THE SIMULATED TIME THAT THE UNIT ENTERS THE  
# 8 NODE AS DETERMINED DURING CONSTRUCTION OF THE SOL-  
# 9 UTION+VECTOR. TIME UNITS ARE MINUTES.  
# 10  
# 11 THIS IS THE SIMULATED TIME THAT THE UNIT LEAVES THE  
# 12 NODE. IT IS COMPUTED AS A COMPANION WITH THE ABOVE  
# 13 WORD #2. THE UNITS ARE IDENTICAL.  
# 14  
# 15 THIS NODE+ARRAY INDEX POINTS TO ANOTHER ENTRY IN  
# 16 THIS SOLUTION VECTOR LIST WHICH IS ASSOCIATED WITH  
# 17 THE SAME NODE+VECTOR. THIS LINK SHOWS THE MULTIPLE USE  
# 18 OF A NODE BY SEVERAL ROUTES OF DIFFERENT UNITS. CHECKING  
# 19 THE TIMES IN AND OUT (WORDS 2 AND 3) OF THE MULTIPLE USERS  
# 20 OF THE NODE WILL INDICATE WHETHER OR NOT A CONFLICT HAS  
# 21 OCCURRED.  
# 22  
# 23 THIS IS THE LANE NUMBER WHICH THIS UNIT USED TO TRAVEL  
# 24 THROUGH THIS NODE. 1-4  
# 25  
# 26 TIME IN MINUTES SPENT PARKING AT THE PREDECESSOR NODE  
# 27 BEFORE TRAVELING TO THIS NODE. +,- 0-32767 MI  
# 28 A POSITIVE VALUE INDICATES THE VECTOR SHOULD BE CON-  
# 29 sidered WHEN DE-CONFLICTING. IF NEGATIVE, IT MUST BE  
# 30 SKIPPED.  
#####  
#  
#####
```

517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546





ROUTE VECTOR FORMAT

#	#	WORD #	CONTENTS	#
#	1	1	UNIT NUMBER	569
#	2	2	LIST HEAD POINTER	570
#	3	3	TOTAL ROUTE TIME	571
#	4	4	TOTAL ROUTE RISK MEASURE	572
#	5	5	TOTAL ROUTE DISTANCE	573
#	6	6	NUMBER OF SOLUTION VECTORS IN ROUTE	574
#	7	7	MOVEMENT TYPE	575
#	8	8	INITIAL TIME	576
#	9	9		577
#	10	10		578



ROUTE VECTOR FORMAT DESCRIPTION

#	WORD #	DESCRIPTION AND USE	RANGE	
# 1	1	INDEX TO UNIT DATA BASE, SO THAT WE KNOW WHICH UNIT USES THIS ROUTE.	1-60	580
# 2	2	INDEX TO THE SOLUTION VECTOR LIST, POINTING TO THE FIRST SOLUTION VECTOR WHICH HAS THIS ROUTE NUMBER.	1-5000	581
# 3	3	TOTAL ELAPSED TIME OF MARCH FROM STARTING NODE TO DESTINATION NODE, INCLUDING RESTS AND STOPS, IN MINUTES.	0-15839 MIN.	582
# 4	4	TOTAL CUMULATIVE RISK MEASURE OF ROUTE	0-32767	583
# 5	5	TOTAL ROUTE DISTANCE, IN TENTHS OF KMS.	0-3276.7	584
# 6	6	TOTAL NUMBER OF NODES VISITED	1-400	585
# 7	7	CODE INDICATING THAT THIS WAS DETERMINED USING A STARTING TIME (=0) OR AN ARRIVAL TIME (=1).	0 OR 1	586
# 8	8	TIME, IN MINUTES, WHICH THE USER DESIGNATED AS EITHER A STARTING TIME OR AN ARRIVAL TIME. DICTATED BY MOVEMENT TYPE.	0-15839	587
# 9	9			588
# 10	10			589
# 11	11			590
# 12	12			591
# 13	13			592
# 14	14			593
# 15	15			594
# 16	16			595
# 17	17			596
# 18	18			597
# 19	19			598
# 20	20			599
# 21	21			600
# 22	22			601
# 23	23			602
# 24	24			603
# 25	25			604
# 26	26			605

# MAP DIRECTORY

THIS DATA BASE KEEPS TRACK OF THE MAPS DATA BASE. THERE IS ONE ENTRY FOR EACH MAP IN THE DISK FILE. THIS ENTRY CONTAINS TWO PIECES OF INFORMATION: 1) MAP NUMBER, AND 2) NUMBER OF NODES IN THE MAP. THE FIRST ENTRY IN THE MAP DIRECTORY CONTAINS A VALUE EQUAL TO THE NUMBER OF MAPS WHICH HAVE BEEN ENTERED. WHEN A MAP'S NODE INFORMATION IS PLACED IN THE DATA BASE, ITS DIRECTORY ENTRY IS PLACED IN THE NEXT AVAILABLE LOCATION OF THE MAP DIRECTORY. THUS THE SECOND MAP DIRECTORY ENTRY CORRESPONDS TO THE FIRST MAP PLACED IN THE MAPS DATA BASE, THE THIRD ENTRY CORRESPONDS TO THE SECOND MAP, ETC.



TRU, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
DATA BASE DESCRIPTIONS

PAGE 24

CROSS-COUNTRY DIRECTORY

#####  
# 1 THIS DIRECTORY IS USED TO FIND SPECIFIC MAPS IN THE CROSS-COUNTRY  
# 2 AND CONCEALMENT DISK FILES. EACH MAP HAS ONE DIRECTORY ENTRY, THIS EN-  
# 3 TRY CONTAINING THE MAP NUMBER, AND THE UTM LATITUDE AND LONGITUDE EX-  
# 4 TREMES OF THE MAP. THESE ARE THE COORDINATES OF THE LOWER LEFT AND UP-  
# 5 PER RIGHT CORNERS. THE FIRST ENTRY OF THE DIRECTORY CONTAINS ONLY A  
# 6 COUNTER WHICH KEEPS TRACK OF HOW MANY ENTRIES ARE PRESENT IN THE DIREC-  
# 7 TORY.  
#####

617  
618  
619  
620  
621  
622  
623

TRM, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM

PAGE 25

\*\*\*\*\*  
\*  
\* PREPARATION FOR USE \*  
\*  
\*\*\*\*\*



PREPARATION

REF	PAGE		
27	626	1	do ONCE BEFORE MAIN-PROGRAM EXECUTION
33	627	2	CREATE NODE<-VECTOR AND MAP<-DIRECTORY FILES ON DISK
	628	3	CREATE CROSS-COUNTRY, CONCEALMENT, AND DIRECTORY FILES ON DISK
	629	4	enddo ONCE BEFORE MAIN-PROGRAM EXECUTION
	630	5	do AS REQUIRED
	631	6	AMEND FILES
	632	7	enddo AS REQUIRED

CREATE NODE+VECTOR AND MAP+DIRECTORY FILES ON DISK

REF	PAGE
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	22
23	23
24	24
25	25
26	26
27	27
28	28
29	29
30	30
31	31
32	32
33	33
34	34
35	35
36	36
37	37
38	38
39	39
40	40
41	41
42	42
43	43
44	44
45	45
46	46
47	47
48	48
49	49
50	50
51	51
52	52
53	53
54	54
55	55
56	56
57	57
58	58
59	59
60	60
61	61
62	62
63	63
64	64
65	65
66	66
67	67
68	68
69	69
70	70
71	71
72	72
73	73
74	74
75	75
76	76
77	77
78	78
79	79
80	80
81	81
82	82
83	83
84	84
85	85
86	86
87	87
88	88
89	89
90	90
91	91
92	92
93	93
94	94
95	95
96	96
97	97
98	98
99	99
100	100

```

1  .BEFORE THE FILES MAY BE CREATED, IT IS NECESSARY TO CREATE
2  .NODE AND LINK CARDS. INFORMATION FOR THESE CARDS IS VISUALLY
3  .DETERMINED FROM THE PARTICULAR L-SERIES MAP BEING CONSIDERED.
4  .
5  .do FOR EACH MAP
6  .do FOR EACH NODE
7  .CREATE NODE CARD
8  .enddo FOR EACH NODE
9  .do FOR EACH LINK
10 .CREATE LINK CARD
11 .enddo FOR EACH LINK
12 .enddo FOR EACH MAP
13 .
14 .NOW THAT THE CARDS ARE READY FOR PROCESSING, CREATE THE NODE+VECTOR
15 .FILE ON DISK, THAT IS, ESTABLISH NODE+VECTORS USING INFORMATION GAINED
16 .FROM NODE AND LINK CARDS.
17 .FIRST, INITIALIZE THE FILES
18 .do ONCE
19 .CREATE DISK FILE FOR NODE+VECTORS, 'MAPS.DAT'
20 .CREATE DISK FILE FOR MAP+DIRECTORY, 'MAPDIR.DAT'
21 .enddo ONCE
22 .
23 .NOW BUILD THE VECTORS IN MEMORY
24 .
25 .do UNTIL ALL MAPS ARE PROCESSED
26 .do FOR EACH MAP
27 .do UNTIL NO MORE NODE CARDS
28 .READ A NODE CARD
29 .PLACE INFORMATION INTO MAIN MEMORY
30 .enddo UNTIL NO MORE NODE CARDS
31 .do UNTIL NO MORE LINK CARDS
32 .

```



TRU, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
PREPARATION FOR USE

PAGE 27.001

```

* 33 READ A LINK CARD... NODE A TO NODE B
* 34 FILL IN LINK INFORMATION TO NODE<VECTOR FOR A
* 35 IF NODE<VECTOR FOR B IS ON THIS MAP
* 36 FILL IN LINK INFORMATION TO NODE<VECTOR FOR B
* 37 ELSE CYCLE
* 38 ENDIF NODE<VECTOR FOR B IS ON THIS MAP
* 39 ENDDO UNTIL NO MORE LINK CARDS
* 40 READ 'MAPDIR.DAT' INTO MEMORY
* 41 READ NUMBER OF MAPS PROCESSED
* 42 INCREMENT NUMBER OF MAPS PROCESSED
* 43 CREATE MAP<DIRECTORY ENTRY FOR THIS MAP
* 44 WRITE NODE<VECTOR ARRAY FOR THIS MAP TO 'MAPS.DAT'
* 45 WRITE MAP<DIRECTORY TO 'MAPDIR.DAT'
* 46 ENDDO FOR EACH MAP
* 47 ENDDO UNTIL ALL MAPS ARE PROCESSED
*
32

```

666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680

TRM, INC. LOCATION AND MOVEMENT ANALYSIS SYSTEM  
21 FEB 78 PREPARATION FOR USE

PAGE 28

CREATE NODE CARD

REF  
PAGE

\* 1 .NODE CARDS WILL BE KEY PUNCHED MANUALLY, USING THE FOLLOWING FORMAT FOR  
\* 2 .THEIR CONSTRUCTION.  
\*

682  
683



NODE CARD FORMAT

#	ENTRY	EXAMPLE	FIELD	#
# 1	NODE MAP NUMBER. THE	4924	1-5(5)	685
# 2	1:50000 MAPS WILL BE USED.			686
# 3	ENTER THE NUMERICAL MAP			687
# 4	NUMBER, NOT THE LETTERS.			688
# 5				689
# 6				690
# 7				691
# 8	NODE NUMBER WITHIN THE MAP.	18	6-9(4)	692
# 9	A MASTER OVERLAY (PLASTIC)			693
# 10	FOR EACH MAP MUST BE KEPT			694
# 11	TO KEEP TRACK OF THE NUMBERS			695
# 12	AND NOTES.			696
# 13				697
# 14	NODE LATITUDE IN KM'S FROM	3245	10-16(7)	698
# 15	THE EQUATOR, READ DIRECTLY			699
# 16	FROM THE 1:50000 MAP.			700
# 17				701
# 18	NODE LONGITUDE IN KM'S FROM	125	17-23(7)	702
# 19	GREENWICH, READ DIRECTLY			703
# 20	FROM THE 1:50000 MAP.			704
# 21				705
# 22	FIRST ADJACENT NODE MAP #.	4923	24-28(5)	706
# 23				707
# 24	FIRST ADJACENT NODE NUMBER	201	29-32(4)	708
# 25	FOR THE MAP OF THIS ADJACENT			709
# 26	NODE. ENTER THE NUMBER ONLY.			710
# 27				711
# 28	SECOND ADJACENT NODE DATA,		33-41(9)	712
# 29	ENTER THE NUMBER ONLY.			713
# 30				714
# 31	THIRD ADJACENT NODE DATA,		42-50(9)	715
# 32	ENTER THE NUMBER ONLY.			716
# 33				717
# 34	FOURTH ADJACENT NODE DATA,		51-59(9)	718
# 35	ENTER THE NUMBER ONLY.			719
# 36				720

TRW, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
PREPARATION FOR USE

PAGE 29.001

# 37	NUMBER OF ADJACENT NODES.	4	60-61(2)	#	721
# 38				#	722
#				#	
#####				#####	



TRM, INC. LOCATION AND MOVEMENT ANALYSIS SYSTEM  
21 FEB 78 PREPARATION FOR USE

PAGE 30

CREATE LINK CARD

REF  
PAGE

1 . LINK CARDS WILL BE KEY-PUNCHED MANUALLY, USING THE  
2 . FOLLOWING FORMAT.

724  
725

TRU, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
PREPARATION FOR USE

PAGE 31

LINK CARD FORMAT

# 1	ENTRY	EXAMPLE	FIELD	#
# 2				#
# 3	FIRST NODE MAP NUMBER.	4924	1-5(5)	#
# 4				#
# 5	FIRST NODE NUMBER WITHIN MAP.	13	6-9(4)	#
# 6				#
# 7	SECOND NODE MAP NUMBER.	4924	10-14(5)	#
# 8				#
# 9	SECOND NODE NUMBER WITHIN MAP.	17	15-18(4)	#
# 10				#
# 11				#
# 12	LINK DISTANCE IN TENTHS OF A	18.7	19-23(5)	#
# 13	KILOMETER BETWEEN FIRST AND			#
# 14	SECOND NODES.			#
# 15				#
# 16	NUMBER OF EQUIVALENT LANES.	3	24-26(3)	#
# 17				#
# 18	ROAD TYPE CODE.	3	27-28(2)	#
# 19				#
# 20	GROUND COVER CODE.	3	29-30(2)	#
# 21				#
# 22	TERRAIN CODE.	2	31-32(2)	#
# 23				#
# 24	BRIDGE CODE.	0	33-34(2)	#
# 25				#
# 26	CITY CODE.	1	35-36(2)	#
#				#



TRM, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
PREPARATION FOR USE

PAGE 32

CREATE MAP<DIRECTORY ENTRY FOR THIS MAP

RET  
PAGE

```
1 .AT THIS POINT IN THE PROGRAM, THIS MAP'S NODE<VECTORS HAVE
2 .BEEN READ INTO CORE, AND ALL PERTINENT INFORMATION HAS BEEN
3 .PLACED IN THE PROPER AREA. THE MAP<NUMBER IS IN THE FIRST
4 .WORD OF THE NODE<VECTOR ARRAY, AND THE NUMBER OF NODES IN
5 .THIS MAP IS KNOWN FROM A COUNTER WHICH WAS INCREMENTED EVERY TIME
6 .A NEW NODE CARD WAS READ.
7
8 MAP<DIRECTORY ENTRY(1,NUMBER OF MAPS)=MAP<NUMBER
9 MAP<DIRECTORY ENTRY(2,NUMBER<OF+MAPS)=NUMBER OF NODES
```

754  
755  
756  
757  
758  
759  
760  
761  
762

TRM, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
PREPARATION FOR USE

PAGE 33

CREATE CROSS-COUNTRY, CONCEALMENT, AND DIRECTORY FILES ON DISK

REF  
PAGE

```
1  .BEFORE THESE THREE FILES MAY BE ESTABLISHED, THE DATA MUST BE
2  .PLACED ON COMPUTER CARDS. THIS DATA IS OBTAINED BY PLACING A
3  .93 X 88 GRID OVER THE APPROPRIATE MAP. EACH SQUARE OF THE GRID
4  .IS ASSIGNED A VALUE EQUAL TO THE CODE WHICH MOST DESCRIBES THE
5  .AREA. THAT IS, IF THE AREA UNDERNEATH THE SQUARE IS PARTIALLY
6  .CODE 4, AND PARTIALLY CODE 5, THE VALUE ASSIGNED IS EITHER 4 OR
7  .5, DEPENDING UPON WHICH APPEARS TO OCCUPY THE MAJORITY OF THE
8  .SQUARE. IF MORE THAN 2 CODES ARE COVERED, THEN THE CODE CHOSEN
9  .IS THE ONE WHICH DOMINATES (SEEMS TO BE GREATER THAN THE OTHERS
10 .TAKEN INDIVIDUALLY) THE SQUARE.
11 .
12 do FOR EACH MAP
13   CREATE DATA CARDS
14   enddo FOR EACH MAP
15 .
16 .WITH THE CARDS ALL MADE, IT IS NOW A MATTER OF READING THE CARDS
17 .INTO MEMORY, CONVERTING THE VALUES AS APPROPRIATE, MAKING A DIR-
18 .ECTORY ENTRY, AND WRITING THE RESULTS OUT TO DISK.
19 .
20 do ONCE
21   CREATE DISK FILE FOR CROSS-COUNTRY MAPS, 'CCM.DAT'
22   CREATE DISK FILE FOR WINTER CONCEALMENT MAPS, 'WCON.DAT'
23   CREATE DISK FILE FOR SUMMER CONCEALMENT MAPS, 'SCON.DAT'
24   CREATE DISK FILE FOR DIRECTORY, 'CONDIR.DAT'
25   enddo ONCE
26 .
27 .BECAUSE THE ALLOWABLE CUBES HAVE THE VALUES 1-7 FOR CROSS-COUN-
28 .TRY AND 8-11 FOR CONCEALMENT, ONLY FOUR BITS ARE NEEDED TO STORE
29 .THIS INFORMATION. A GOOD SPACE SAVING MAY BE MADE IF THE VALUES
30 .ARE PACKED FOUR TO A WORD. DOING THIS, THOUGH, MEANS THAT, IN
31 .ORDER TO DEAL WITH COMPLETE ROWS OF DATA, FOUR ROWS OF CARDS
32 .SHOULD BE PROCESSED AT ONE TIME.
```

764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795





TRM, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM

PAGE 34

\*\*\*\*\*  
\* LAMAS FLEW \*  
\*\*\*\*\*



LAMAS

REF  
PAGE

```

* 1 do UPON MAIN+PROGRAM EXECUTION
* 2 do FOREVER
* 3 DISPLAY MAIN FUNCTION MENU
* 4 WAIT FOR USER INPUT
* 5 do CASE OF :
* 6 ROAD+NETWORK:
* 7 IMPLEMENT ALGORITHMS AND FUNCTIONS USING THE NODE+VECTOR DATA BASE
36 * 8 TERRAIN+MODEL:
63 * 9 IMPLEMENT ALGORITHMS AND FUNCTIONS USING THE CROSS-COUNTRY DATA BASE
* 10 CONCEALMENT DATA BASES
* 11 EXIT:
* 12 EXIT THIS SYSTEM
* 13 enddo CASE OF
* 14 enddo FOREVER
* 15 enddo UPON MAIN+PROGRAM EXECUTION
*
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841

```

IMPLEMENT ALGORITHMS AND FUNCTIONS USING THE NODE-VECTOR DATA BASE

REF  
PAGE

```

1  do UPON FUNCTION REQUEST
2  do FOREVER
3  DISPLAY FUNCTION MENU
4  WAIT FOR USER INPUT
5  do CASE OF
6  INITIALIZE:
7  PREPARE FOR ROAD NETWORK PATH CALCULATIONS
8  CALCULATION:
9  PATH DETERMINATION AND DISPLAY
10 EXT:
11 return TO CALLING PROGRAM
12 enddo CASE OF
13 enddo FOREVER
14 enddo UPON FUNCTION REQUEST

```

843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856



TRW, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
LAMAS FLON

PAGE 37

PREPARE FOR ROAD NETWORK PATH CALCULATIONS

REF  
PAGE

```

1  do UPON FUNCTION REQUEST
2  do FOREVER
3  DISPLAY FUNCTION MENU
4  WAIT FOR USER INPUT
5  do CASE OF
6  MAPS:
7  UNITS:
8  UNITS:
9  UNITS:
10 FERG:
11 FERG:
12 EXIT:
13
14
15
16  enddo UPON FUNCTION REQUEST
17
18  return TO CALLING PROGRAM
19  enddo CASE OF
20  enddo FOREVER
21  enddo UPON FUNCTION REQUEST
22
23  ESTABLISH A FORWARD EDGE OF BATTLE AREA
24
25  INITIALIZE UNIT VECTORS
26
27  INITIALIZE MAP NUMBERS AND SCREEN EXTREMES
28
29  do CASE OF
30  MAPS:
31  UNITS:
32  UNITS:
33  UNITS:
34  UNITS:
35  UNITS:
36  UNITS:
37  UNITS:
38  UNITS:
39  UNITS:
40  UNITS:
41  UNITS:
42  UNITS:
43  UNITS:
44  UNITS:
45  UNITS:
46  UNITS:
47  UNITS:
48  UNITS:
49  UNITS:
50  UNITS:
51  UNITS:
52  UNITS:
53  UNITS:
54  UNITS:
55  UNITS:
56  UNITS:
57  UNITS:
58  UNITS:
59  UNITS:
60  UNITS:
61  UNITS:
62  UNITS:
63  UNITS:
64  UNITS:
65  UNITS:
66  UNITS:
67  UNITS:
68  UNITS:
69  UNITS:
70  UNITS:
71  UNITS:
72  UNITS:
73  UNITS:
74  UNITS:
75  UNITS:
76  UNITS:
77  UNITS:
78  UNITS:
79  UNITS:
80  UNITS:
81  UNITS:
82  UNITS:
83  UNITS:
84  UNITS:
85  UNITS:
86  UNITS:
87  UNITS:
88  UNITS:
89  UNITS:
90  UNITS:
91  UNITS:
92  UNITS:
93  UNITS:
94  UNITS:
95  UNITS:
96  UNITS:
97  UNITS:
98  UNITS:
99  UNITS:
100 UNITS:

```

858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873

INITIALIZE MAP+NUMBERS AND SCREEN EXTREMES

REF

PAGE

```

1  do UPON FUNCTION REQUEST
2  ESTABLISH DIRECTORY IN MEMORY AND READ IN NODE+VECTORS
3  do FOR EACH NODE IN MEMORY
4  do FOR EACH ADJACENT NODE
5  if ADJACENT NODE'S MAP+NUMBER IS IN THE DIRECTORY
6  CHANGE ADJACENT NODE'S MAP+NUMBER TO INDEX
7  endif ADJACENT NODE'S MAP+NUMBER
8  enddo FOR EACH ADJACENT NODE
9  enddo FOR EACH NODE IN MEMORY
10 PROMPT FOR ENTRY OF SCREEN EXTREMES OF LATITUDE AND LONGITUDE
11 READ INPUT
12 PLACE INPUT IN COMMON STORAGE
13 return TO CALLING PROGRAM
14 enddo UPON FUNCTION REQUEST

```

875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888



ESTABLISH DIRECTORY IN MEMORY AND READ IN NODE+VECTORS

REF  
PAGE

```

1  . THIS DIRECTORY CONSISTS OF A NUMBER OF ENTRIES, ONE FOR EACH
2  . MAP WHICH THE USER WISHES TO EMPLOY. THE FIRST DIRECTORY ENTRY CORRE-
3  . SPONDS TO THE FIRST MAP+NUMBER REQUESTED BY THE USER, AND SO ON FOR
4  . EACH ENTRY. THERE ARE THREE ELEMENTS IN EACH ENTRY, 1) THE MAP+NUMBER,
5  . 2) THE NUMBER OF NODES IN THE MAP, AND 3) THE NUMBER OF NODES ALREADY
6  . PRESENT IN MEMORY.
7  . GIVEN THIS MAP+DIRECTORY, IT IS A SIMPLE MATTER TO READ A MAP'S
8  . NODES INTO MAIN MEMORY, SINCE WE KNOW HOW MANY TO READ IN,
9  . MAP+DIRECTORY WORD 2, AND WHERE TO START PUTTING THEM, MAP+DIRECTORY
10 . WORD 3 +1.
11 .
12 . READ DISK FILE MAP+DIRECTORY INTO MAIN MEMORY
13 . PROMPT FOR ENTRY OF MAP+NUMBERS
14 . READ USER INPUT OF MAP+NUMBERS TO BE PROCESSED
15 . DO FOR EACH MAP+NUMBER ENTERED BY USER
16 .   IF MAP+NUMBER IS IN THE DISK FILE MAP+DIRECTORY...AS THE 1 TH ENTRY
17 .     SET FIRST WORD OF MAIN MEMORY MAP+DIRECTORY TO 1
18 .   SET SECOND WORD TO NUMBER OF NODES IN THIS MAP
19 .   SET THIRD WORD TO NUMBER OF NODES ALREADY PRESENT IN MEMORY
20 .   READ THIS MAP'S NODES INTO MAIN MEMORY
21 .   RESET FIRST WORD TO THIS MAP+NUMBER
22 .   ELSE MAP+NUMBER IS NOT IN THE DATA BASE
23 .     PRINT ERROR MESSAGE
24 .     LET THE USER RE-ENTER THE MAP+NUMBER
25 .   cycle
26 .   end if MAP+NUMBER IS IN THE DISK FILE MAP+DIRECTORY
27 .   end do FOR EACH MAP+NUMBER ENTERED BY USER

```

40

READ THIS MAP'S NODES INTO MAIN MEMORY

REF  
PAGE

```
* 1 . AT THIS POINT IN THE PROGRAM, THE MAP+DIRECTORY IN MAIN MEMORY
* 2 . HAS ITS AS ELEMENTS 1) THE INDEX 1, INDICATING THAT THIS MAP IS THE
* 3 . 1 TH ONE IN THE DISK FILE MAP+DIRECTORY, 2) NUMBER OF NODES IN THIS MAP
* 4 . AND 3) NUMBER OF NODES ALREADY PRESENT IN MEMORY. IN ORDER TO
* 5 . READ THIS MAP'S NODES INTO MEMORY SO THAT NO SPACE IS WASTED,
* 6 . WE NEED ONLY CONSIDER THE INFORMATION JUST LISTED. TO READ DATA
* 7 . FROM DISK TO MEMORY WE NEED TO KNOW HOW MANY BYTES WILL BE MOVED.
* 8 . THIS IS JUST WORD 2 MULTIPLIED BY 64 (32 WORDS * 2 BYTES PER WORD). WE
* 9 . ALSO MUST KNOW EXACTLY WHERE THE DATA WILL BE PLACED IN MEMORY, THAT
* 10 . IS, THE NEXT AVAILABLE SPACE IN THE NODE+VECTOR ARRAY.
* 11 . THIS IS SIMPLY WORD 3 PLUS ONE. FINALLY, WE MUST KNOW FROM WHAT
* 12 . DISK BLOCK IS THE DATA COMING. THIS IS 1-1 MULTIPLIED BY 10, SINCE
* 13 . THE FIRST MAP STARTS AT BLOCK ONE, THE SECOND AT BLOCK 11, ETC.
* 14 . NOTE THAT NO MAP HAS MORE THAN 80 NODE+VECTORS, EACH VECTOR
* 15 . HAVING 32 WORDS, SO THERE ARE 8 NODE+VECTORS TO A DISK BLOCK OF
* 16 . 256 WORDS.
* 17 .
* 18 . BYTES = (MAP+DIRECTORY WORD 2) * 64
* 19 . INDEX = (MAP+DIRECTORY WORD 3) + 1
* 20 . STARTING BLOCK = ((MAP+DIRECTORY WORD 1) - 1) * 10
* 21 . PERFORM DISK READ (BYTES, INDEX, STARTING BLOCK)
```

918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938



TRM, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
LAMAS FLOBI

PAGE 41

CHOOSE ADJACENT NODE'S MAP+NUMBER TO INDEX

REF  
PAGE

1 . IN ORDER TO ACCESS A NODE QUICKLY, IT IS NECESSARY THAT ONE NOT  
2 . HAVE TO USE THE MAP+DIRECTORY EACH TIME A REFERENCE IS MADE.  
3 . THUS, CONSIDER THE NODE+VECTOR ARRAY AS A LIST OF NODES,  
4 . INDEXABLE BY INTEGERS, SUCH THAT THE FIRST NODE+VECTOR HAS INDEX=1,  
5 . THE SECOND HAS INDEX=2, AND SO ON FOR THE ENTIRE LIST. THEN, IF  
6 . WE KNOW THE MAP+NUMBER AND THE NODE+NUMBER OF A PARTICULAR NODE+VECTOR,  
7 . THEN THIS INDEX MAY BE CALCULATED ONCE, AND SAVED. THE MOST LOGICAL  
8 . PLACE TO SAVE IT IS IN IT'S ADJACENT MAP+NUMBER LOCATION.  
9 . THIS CALCULATION IS SIMPLE, AND DEPENDS ON OUR KNOWLEDGE OF JUST  
10 . TWO THINGS, 1) FOR THE ADJACENT NODE'S MAP+NUMBER, THE NUMBER OF NODES  
11 . ALREADY IN CORE, AND 2) FOR THE ADJACENT NODE, ITS NODE+NUMBER. BOTH  
12 . OF THESE PIECES OF INFORMATION ARE READILY AVAILABLE, THE FIRST IS JUST  
13 . THE THIRD WORD OF THE ADJACENT NODE'S MAP+DIRECTORY ENTRY, AND  
14 . THE SECOND IS IN THE NODE+VECTOR.

ADJACENT MAP+NUMBER = (ADJACENT NODE'S MAP+DIRECTORY WORD 3) + ADJACENT NODE'S

940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955

## INITIALIZE UNIT VECTORS

3510d  
432

```

1  . THIS DATA BASE IS COMPLETELY USER DEFINED AT EXECUTION TIME.
2  . THE USER SPECIFIES UNIT NAME, UNIT STARTING AND DESTINATION LO-
3  . CATIONS, UNIT PRIORITY, AND UNIT TYPE CODE.
4
5  . DO UPON FUNCTION REQUEST
6  . ASK IF USER WISHES TO REINITIALIZE OR ADD ON TO EXISTING STRUCTURE
7  . IF USER WANTS TO REINITIALIZE
8  . ERASE ALL UNIT+VECTORS
9  . SET UNIT+COUNTER TO ZERO
10 . endif USER WANTS TO REINITIALIZE
11
12 . . PERFORM THE FOLLOWING AS MANY AS SIXTY TIMES
13 .
14 . DO UNTIL UNIT+COUNTER EQUALS SIXTY
15 .
16 . . PROMPT FOR THE FOLLOWING: UNIT NAME, STARTING LOCATION, DESTINATION,
17 . . PRIORITY, AND TYPE CODE. ERROR CHECKING IS BE PERFORMED ON ALL INPUT
18 . . SAVE THE UNIT NAME. INPUT FOR THE START AND FINISH LOCATIONS IS IN THE
19 . . FORM MAP+NUMBER, NODE+NUMBER. ANY TIME THERE IS AN ERROR, THE PROMPT
20 . . WILL BE REPEATED UNTIL VALID INPUT IS RECEIVED.
21 .
22 . PROMPT USER TO ENTER UNIT NAME
23 . IF UNIT NAME IS NULL
24 .   RETURN TO CALLING PROGRAM
25 . ELSE INPUT IS VALID CHARACTERS
26 .   INCREMENT UNIT+COUNTER
27 .   ASSIGN INPUT TO FIRST FOUR WORDS OF NEXT AVAILABLE UNIT+VECTOR
28 .   PROMPT FOR STARTING LOCATION
29 .   IF STARTING LOCATION IS INVALID
30 .     PRINT ERROR MESSAGE
31 .     TRY AGAIN
32 .   ELSE INPUT IS VALID

```



TRM, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
LAMAS FLOW

PAGE 42.001

```
* 33      ASSIGN INPUT TO UNIT+VECTOR WORD 5 (STARTING LOCATION)
* 34      ASSIGN INPUT TO UNIT+VECTOR WORD 7 (PRESENT LOCATION)
* 35      PROMPT FOR DESTINATION
* 36      IF DESTINATION IS INVALID
* 37          PRINT ERROR MESSAGE
* 38          TRY AGAIN
* 39      ELSE INPUT IS VALID
* 40          ASSIGN INPUT TO UNIT+VECTOR WORD 6 (DESTINATION)
* 41          PROMPT FOR PRIORITY
* 42          IF PRIORITY IS INVALID
* 43              PRINT ERROR MESSAGE
* 44              TRY AGAIN
* 45          ELSE INPUT IS VALID
* 46              ASSIGN INPUT TO FIRST BYTE OF UNIT+VECTOR WORD 8
* 47              PROMPT FOR TYPE CODE
* 48              IF TYPE CODE IS INVALID
* 49                  PRINT ERROR MESSAGE
* 50                  TRY AGAIN
* 51              ELSE INPUT IS VALID
* 52                  ASSIGN INPUT TO SECOND BYTE OF UNIT+VECTOR WORD 8
* 53                  ENDIF TYPE CODE IS INVALID
* 54                  ENDIF PRIORITY IS INVALID
* 55                  ENDIF DESTINATION IS INVALID
* 56                  ENDIF STARTING LOCATION IS INVALID
* 57                  ENDIF UNIT NAME IS NULL
* 58                  ENDDO UNTIL UNIT+COUNTER EQUALS SIXTY
* 59                  ENDDO UPON FUNCTION REQUEST
```

990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016

ESTABLISH A FORWARD EDGE OF BATTLE AREA

REF  
PAGE

```

1  .AN ARBITRARY MAXIMUM OF SIX NODES MAY BE DESIGNATED AS FEBA POINTS.
2  .
3  .do UPON FUNCTION REQUEST
4  .do SIX TIMES
5  .  PROMPT FOR MAP+NUMBER AND NODE+NUMBER, OR EXIT
6  .  if INPUT IS EXIT
7  .    return TO CALLING PROGRAM
8  .  endif INPUT IS EXIT
9  .  CHECK MAP+NUMBER AND NODE+NUMBER FOR LEGITIMACY
10 .  if NOT VALID
11 .    PRINT ERROR MESSAGE
12 .    TRY AGAIN
13 .  else VALID
14 .    FIND NODE+VECTOR CORRESPONDING TO MAP+NUMBER, NODE+NUMBER
15 .    ASSIGN NODE+VECTOR TO FEBA ARRAY
16 .  endif NOT VALID
17 .  enddo SIX TIMES
18 .  enddo UPON FUNCTION REQUEST

```

1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035



FIND NODE+VECTOR NEAREST TO UTM COORDINATES

KEY  
PAGE

```

*****
1  . THIS WILL BE ACCOMPLISHED BY STARTING AT THE BEGINNING OF THE
2  . NODE+VECTOR ARRAY AND COMPUTING EACH NODE+VECTOR 'S DISTANCE FROM THE
3  . UTM COORDINATES USING A LEAST SQUARES APPROACH. THAT IS, DISTANCE IS
4  . CALCULATED TO BE THE SUM OF THE SQUARES OF THE DIFFERENCES BETWEEN THE
5  . NODE+VECTOR 'S LATITUDE AND LONGITUDE AND THE CALCULATED UTM COORDI-
6  . NATES.
7
8  . LEAST = INFINITY
9  . DO FOR EACH NODE+VECTOR
10 .   DISTANCE = (UTM LAT. - 'LAT.) ** 2 + (UTM LON. - LON.) ** 2
11 .   IF DISTANCE IS LESS THAN LEAST
12 .     REMEMBER THE NODE+VECTOR
13 .     LEAST = DISTANCE
14 .   endif DISTANCE IS LESS THAN LEAST
15 . enddo FOR EACH NODE+VECTOR
16
17 . REMEMBERED NODE+VECTOR IS THE ONE WE WANT
18
*****
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054

```

## PATH DETERMINATION AND DISPLAY

REF

PAGE

```
*****
* 1 .THESE ARE THE FUNCTIONS WHICH PERFORM THE FUNDAMENTAL MOVEMENT ALGO-
* 2 .RITHM. THE MAIN PURPOSE IS TO COMPUTE OPTIMAL PATHS FOR DEFINED
* 3 .GROUND FORCE UNITS, SO THAT CERTAIN CRITERIA IS SATISFIED. THE GENERAL
* 4 .PROCEDURE IS, GIVEN A UNIT'S STARTING NODE, DESTINATION NODE, AND
* 5 .DESIRED ARRIVAL TIME AT DESTINATION, OR START TIME AT THE BEGINNING,
* 6 .CALCULATE THE BEST PATH FOR THE UNIT TO TRAVEL. 'BEST' IS USER-DEFINED
* 7 .AS A FUNCTION OF TIME AND RISK. THE FORMULA USED IS K*TIME + C*RISK,
* 8 .WHERE K AND C ARE CONSTANTS, C EQUALING ZERO OR ONE, AND K BEING BE-
* 9 .WEEN ZERO AND TEN. BOTH K AND C MAY NOT BE ZERO AT THE SAME TIME.
* 10 .USING THE BASIC PATH ALGORITHM, CERTAIN VARIATIONS OF THE PATH CALCU-
* 11 .LATION MAY BE PERFORMED, SUCH AS FINDING THE SECOND SHORTEST PATH,
* 12 .FINDING THE OPTIMAL NODE FOR INTERDICTION, AND, GIVEN A NUMBER OF UNITS
* 13 .TRAVELING THROUGH A ROAD NETWORK, FIND WHICH ORDER OF PRIORITIES OPTI-
* 14 .MIZES THE ENTIRE NETWORKS USAGE.
* 15 . THERE ARE TWO MORE OPTIONS FROM WHICH THE USER MAY CHOOSE, INTERDIC-
* 16 .TION AND SOLUTIONS. THEIR PROPERTIES WILL BE EXPLAINED LATER.
* 17
* 18 . do UPON FUNCTION REQUEST
* 19 . do FOREVER
* 20 . DISPLAY FUNCTION MENU
* 21 . WAIT FOR USER INPUT
* 22 . do CASE OF
* 23 INTERDICTION:
* 24 46
* 25 53
* 26 60
* 27 61
* 28 62
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 38
* 39
* 40
* 41
* 42
* 43
* 44
* 45
* 46
* 47
* 48
* 49
* 50
* 51
* 52
* 53
* 54
* 55
* 56
* 57
* 58
* 59
* 60
* 61
* 62
* 63
* 64
* 65
* 66
* 67
* 68
* 69
* 70
* 71
* 72
* 73
* 74
* 75
* 76
* 77
* 78
* 79
* 80
* 81
* 82
* 83
* 84
* 85
* 86
* 87
* 88
* 89
* 90
* 91
* 92
* 93
* 94
* 95
* 96
* 97
* 98
* 99
* 100
* 101
* 102
* 103
* 104
* 105
* 106
* 107
* 108
* 109
* 110
* 111
* 112
* 113
* 114
* 115
* 116
* 117
* 118
* 119
* 120
* 121
* 122
* 123
* 124
* 125
* 126
* 127
* 128
* 129
* 130
* 131
* 132
* 133
* 134
* 135
* 136
* 137
* 138
* 139
* 140
* 141
* 142
* 143
* 144
* 145
* 146
* 147
* 148
* 149
* 150
* 151
* 152
* 153
* 154
* 155
* 156
* 157
* 158
* 159
* 160
* 161
* 162
* 163
* 164
* 165
* 166
* 167
* 168
* 169
* 170
* 171
* 172
* 173
* 174
* 175
* 176
* 177
* 178
* 179
* 180
* 181
* 182
* 183
* 184
* 185
* 186
* 187
* 188
* 189
* 190
* 191
* 192
* 193
* 194
* 195
* 196
* 197
* 198
* 199
* 200
* 201
* 202
* 203
* 204
* 205
* 206
* 207
* 208
* 209
* 210
* 211
* 212
* 213
* 214
* 215
* 216
* 217
* 218
* 219
* 220
* 221
* 222
* 223
* 224
* 225
* 226
* 227
* 228
* 229
* 230
* 231
* 232
* 233
* 234
* 235
* 236
* 237
* 238
* 239
* 240
* 241
* 242
* 243
* 244
* 245
* 246
* 247
* 248
* 249
* 250
* 251
* 252
* 253
* 254
* 255
* 256
* 257
* 258
* 259
* 260
* 261
* 262
* 263
* 264
* 265
* 266
* 267
* 268
* 269
* 270
* 271
* 272
* 273
* 274
* 275
* 276
* 277
* 278
* 279
* 280
* 281
* 282
* 283
* 284
* 285
* 286
* 287
* 288
* 289
* 290
* 291
* 292
* 293
* 294
* 295
* 296
* 297
* 298
* 299
* 300
* 301
* 302
* 303
* 304
* 305
* 306
* 307
* 308
* 309
* 310
* 311
* 312
* 313
* 314
* 315
* 316
* 317
* 318
* 319
* 320
* 321
* 322
* 323
* 324
* 325
* 326
* 327
* 328
* 329
* 330
* 331
* 332
* 333
* 334
* 335
* 336
* 337
* 338
* 339
* 340
* 341
* 342
* 343
* 344
* 345
* 346
* 347
* 348
* 349
* 350
* 351
* 352
* 353
* 354
* 355
* 356
* 357
* 358
* 359
* 360
* 361
* 362
* 363
* 364
* 365
* 366
* 367
* 368
* 369
* 370
* 371
* 372
* 373
* 374
* 375
* 376
* 377
* 378
* 379
* 380
* 381
* 382
* 383
* 384
* 385
* 386
* 387
* 388
* 389
* 390
* 391
* 392
* 393
* 394
* 395
* 396
* 397
* 398
* 399
* 400
* 401
* 402
* 403
* 404
* 405
* 406
* 407
* 408
* 409
* 410
* 411
* 412
* 413
* 414
* 415
* 416
* 417
* 418
* 419
* 420
* 421
* 422
* 423
* 424
* 425
* 426
* 427
* 428
* 429
* 430
* 431
* 432
* 433
* 434
* 435
* 436
* 437
* 438
* 439
* 440
* 441
* 442
* 443
* 444
* 445
* 446
* 447
* 448
* 449
* 450
* 451
* 452
* 453
* 454
* 455
* 456
* 457
* 458
* 459
* 460
* 461
* 462
* 463
* 464
* 465
* 466
* 467
* 468
* 469
* 470
* 471
* 472
* 473
* 474
* 475
* 476
* 477
* 478
* 479
* 480
* 481
* 482
* 483
* 484
* 485
* 486
* 487
* 488
* 489
* 490
* 491
* 492
* 493
* 494
* 495
* 496
* 497
* 498
* 499
* 500
* 501
* 502
* 503
* 504
* 505
* 506
* 507
* 508
* 509
* 510
* 511
* 512
* 513
* 514
* 515
* 516
* 517
* 518
* 519
* 520
* 521
* 522
* 523
* 524
* 525
* 526
* 527
* 528
* 529
* 530
* 531
* 532
* 533
* 534
* 535
* 536
* 537
* 538
* 539
* 540
* 541
* 542
* 543
* 544
* 545
* 546
* 547
* 548
* 549
* 550
* 551
* 552
* 553
* 554
* 555
* 556
* 557
* 558
* 559
* 560
* 561
* 562
* 563
* 564
* 565
* 566
* 567
* 568
* 569
* 570
* 571
* 572
* 573
* 574
* 575
* 576
* 577
* 578
* 579
* 580
* 581
* 582
* 583
* 584
* 585
* 586
* 587
* 588
* 589
* 590
* 591
* 592
* 593
* 594
* 595
* 596
* 597
* 598
* 599
* 600
* 601
* 602
* 603
* 604
* 605
* 606
* 607
* 608
* 609
* 610
* 611
* 612
* 613
* 614
* 615
* 616
* 617
* 618
* 619
* 620
* 621
* 622
* 623
* 624
* 625
* 626
* 627
* 628
* 629
* 630
* 631
* 632
* 633
* 634
* 635
* 636
* 637
* 638
* 639
* 640
* 641
* 642
* 643
* 644
* 645
* 646
* 647
* 648
* 649
* 650
* 651
* 652
* 653
* 654
* 655
* 656
* 657
* 658
* 659
* 660
* 661
* 662
* 663
* 664
* 665
* 666
* 667
* 668
* 669
* 670
* 671
* 672
* 673
* 674
* 675
* 676
* 677
* 678
* 679
* 680
* 681
* 682
* 683
* 684
* 685
* 686
* 687
* 688
* 689
* 690
* 691
* 692
* 693
* 694
* 695
* 696
* 697
* 698
* 699
* 700
* 701
* 702
* 703
* 704
* 705
* 706
* 707
* 708
* 709
* 710
* 711
* 712
* 713
* 714
* 715
* 716
* 717
* 718
* 719
* 720
* 721
* 722
* 723
* 724
* 725
* 726
* 727
* 728
* 729
* 730
* 731
* 732
* 733
* 734
* 735
* 736
* 737
* 738
* 739
* 740
* 741
* 742
* 743
* 744
* 745
* 746
* 747
* 748
* 749
* 750
* 751
* 752
* 753
* 754
* 755
* 756
* 757
* 758
* 759
* 760
* 761
* 762
* 763
* 764
* 765
* 766
* 767
* 768
* 769
* 770
* 771
* 772
* 773
* 774
* 775
* 776
* 777
* 778
* 779
* 780
* 781
* 782
* 783
* 784
* 785
* 786
* 787
* 788
* 789
* 790
* 791
* 792
* 793
* 794
* 795
* 796
* 797
* 798
* 799
* 800
* 801
* 802
* 803
* 804
* 805
* 806
* 807
* 808
* 809
* 810
* 811
* 812
* 813
* 814
* 815
* 816
* 817
* 818
* 819
* 820
* 821
* 822
* 823
* 824
* 825
* 826
* 827
* 828
* 829
* 830
* 831
* 832
* 833
* 834
* 835
* 836
* 837
* 838
* 839
* 840
* 841
* 842
* 843
* 844
* 845
* 846
* 847
* 848
* 849
* 850
* 851
* 852
* 853
* 854
* 855
* 856
* 857
* 858
* 859
* 860
* 861
* 862
* 863
* 864
* 865
* 866
* 867
* 868
* 869
* 870
* 871
* 872
* 873
* 874
* 875
* 876
* 877
* 878
* 879
* 880
* 881
* 882
* 883
* 884
* 885
* 886
* 887
* 888
* 889
* 890
* 891
* 892
* 893
* 894
* 895
* 896
* 897
* 898
* 899
* 900
* 901
* 902
* 903
* 904
* 905
* 906
* 907
* 908
* 909
* 910
* 911
* 912
* 913
* 914
* 915
* 916
* 917
* 918
* 919
* 920
* 921
* 922
* 923
* 924
* 925
* 926
* 927
* 928
* 929
* 930
* 931
* 932
* 933
* 934
* 935
* 936
* 937
* 938
* 939
* 940
* 941
* 942
* 943
* 944
* 945
* 946
* 947
* 948
* 949
* 950
* 951
* 952
* 953
* 954
* 955
* 956
* 957
* 958
* 959
* 960
* 961
* 962
* 963
* 964
* 965
* 966
* 967
* 968
* 969
* 970
* 971
* 972
* 973
* 974
* 975
* 976
* 977
* 978
* 979
* 980
* 981
* 982
* 983
* 984
* 985
* 986
* 987
* 988
* 989
* 990
* 991
* 992
* 993
* 994
* 995
* 996
* 997
* 998
* 999
* 1000
* 1001
* 1002
* 1003
* 1004
* 1005
* 1006
* 1007
* 1008
* 1009
* 1010
* 1011
* 1012
* 1013
* 1014
* 1015
* 1016
* 1017
* 1018
* 1019
* 1020
* 1021
* 1022
* 1023
* 1024
* 1025
* 1026
* 1027
* 1028
* 1029
* 1030
* 1031
* 1032
* 1033
* 1034
* 1035
* 1036
* 1037
* 1038
* 1039
* 1040
* 1041
* 1042
* 1043
* 1044
* 1045
* 1046
* 1047
* 1048
* 1049
* 1050
* 1051
* 1052
* 1053
* 1054
* 1055
* 1056
* 1057
* 1058
* 1059
* 1060
* 1061
* 1062
* 1063
* 1064
* 1065
* 1066
* 1067
* 1068
* 1069
* 1070
* 1071
* 1072
* 1073
* 1074
* 1075
* 1076
* 1077
* 1078
* 1079
* 1080
* 1081
* 1082
* 1083
* 1084
* 1085
* 1086
* 1087
* 1088
* 1089
* 1090
* 1091
* 1092
* 1093
* 1094
* 1095
* 1096
* 1097
* 1098
* 1099
* 1100
* 1101
* 1102
* 1103
* 1104
* 1105
* 1106
* 1107
* 1108
* 1109
* 1110
* 1111
* 1112
* 1113
* 1114
* 1115
* 1116
* 1117
* 1118
* 1119
* 1120
* 1121
* 1122
* 1123
* 1124
* 1125
* 1126
* 1127
* 1128
* 1129
* 1130
* 1131
* 1132
* 1133
* 1134
* 1135
* 1136
* 1137
* 1138
* 1139
* 1140
* 1141
* 1142
* 1143
* 1144
* 1145
* 1146
* 1147
* 1148
* 1149
* 1150
* 1151
* 1152
* 1153
* 1154
* 1155
* 1156
* 1157
* 1158
* 1159
* 1160
* 1161
* 1162
* 1163
* 1164
* 1165
* 1166
* 1167
* 1168
* 1169
* 1170
* 1171
* 1172
* 1173
* 1174
* 1175
* 1176
* 1177
* 1178
* 1179
* 1180
* 1181
* 1182
* 1183
* 1184
* 1185
* 1186
* 1187
* 1188
* 1189
* 1190
* 1191
* 1192
* 1193
* 1194
* 1195
* 1196
* 1197
* 1198
* 1199
* 1200
* 1201
* 1202
* 1203
* 1204
* 1205
* 1206
* 1207
* 1208
* 1209
* 1210
* 1211
* 1212
* 1213
* 1214
* 1215
* 1216
* 1217
* 1218
* 1219
* 1220
* 1221
* 1222
* 1223
* 1224
* 1225
* 1226
* 1227
* 1228
* 1229
* 1230
* 1231
* 1232
* 1233
* 1234
* 1235
* 1236
* 1237
* 1238
* 1239
* 1240
* 1241
* 1242
* 1243
* 1244
* 1245
* 1246
* 1247
* 1248
* 1249
* 1250
* 1251
* 1252
* 1253
* 1254
* 1255
* 1256
* 1257
* 1258
* 1259
* 1260
* 1261
* 1262
* 1263
* 1264
* 1265
* 1266
* 1267
* 1268
* 1269
* 1270
* 1271
* 1272
* 1273
* 1274
* 1275
* 1276
* 1277
* 1278
* 1279
* 1280
* 1281
* 1282
* 1283
* 1284
* 1285
* 1286
* 1287
* 1288
* 1289
* 1290
* 1291
* 1292
* 1293
* 1294
* 1295
* 1296
* 1297
* 1298
* 1299
* 1300
* 1301
* 1302
* 1303
* 1304
* 1305
* 1306
* 1307
* 1308
* 1309
* 1310
* 1311
* 1312
* 1313
* 1314
* 1315
* 1316
* 1317
* 1318
* 1319
* 1320
* 1321
* 1322
* 1323
* 1324
* 1325
* 1326
* 1327
* 1328
* 1329
* 1330
* 1331
* 1332
* 1333
* 1334
* 1335
* 1336
* 1337
* 1338
* 1339
* 1340
* 1341
* 1342
* 1343
* 1344
* 1345
* 1346
* 1347
* 1348
* 1349
* 1350
* 1351
* 1352
* 1353
* 1354
* 1355
* 1356
* 1357
* 1358
* 1359
* 1360
* 1361
* 1362
* 1363
* 1364
* 1365
* 1366
* 1367
* 1368
* 1369
* 1370
* 1371
* 1372
* 1373
* 1374
* 1375
* 1376
* 1377
* 1378
* 1379
* 1380
* 1381
* 1382
* 1383
* 1384
* 1385
* 1386
* 1387
* 1388
* 1389
* 1390
* 1391
* 1392
* 1393
* 1394
* 1395
* 1396
* 1397
* 1398
* 1399
* 1400
* 1401
* 1402
* 1403
* 1404
* 1405
* 1406
* 1407
* 1408
* 1409
* 1410
* 1411
* 1412
* 1413
* 1414
* 1415
* 1416
* 1417
* 1418
* 1419
* 1420
* 1421
* 1422
* 1423
* 1424
* 1425
* 1426
* 1427
* 1428
* 1429
* 1430
* 1431
* 1432
* 1433
* 1434
* 1435
* 1436
* 1437
* 1438
* 1439
* 1440
* 1441
* 1442
* 1443
* 1444
* 1445
* 1446
* 1447
* 1448
* 1449
* 1450
* 1451
* 1452
* 1453
* 1454
* 1455
* 1456
* 1457
* 1458
* 1459
* 1460
* 1461
* 1462
* 1463
* 1464
* 1465
* 1466
* 1467
* 1468
* 1469
* 1470
* 1471
* 1472
* 1473
* 1474
* 1475
* 1476
* 1477
* 1478
* 1479
* 1480
* 1481
* 1482
* 1483
* 1484
* 1485
* 1486
* 1487
* 1488
* 1489
* 1490
* 1491
* 1492
* 1493
* 1494
* 1495
* 1496
* 1497
* 1498
* 1499
* 1500
* 1501
* 1502
* 1503
* 1504
* 1505
* 1506
* 1507
* 1508
* 1509
* 1510
* 1511
* 1512
* 1513
* 1514
* 1515
* 1516
* 1517
* 1518
* 1519
* 1520
* 1521
* 1522
* 1523
* 1524
* 1525
* 1526
* 1527
* 1528
* 1529
* 1530
* 1531
* 1532
* 1533
* 1534
* 1535
* 1536
* 1537
* 1538
* 1539
* 1540
* 1541
* 1542
* 1543
* 1544
* 1545
* 1546
* 1547
* 1548
* 1549
* 1550
* 1551
* 1552
* 1553
* 1554
* 1555
* 1556
* 1557
* 1558
* 1559
* 1560
* 1561
* 1562
* 1563
* 1564
* 1565
* 1566
* 1567
* 1568
* 1569
* 1570
* 1571
* 1572
* 1573
* 1574
* 1575
* 1576
* 1577
* 1578
* 1579
* 1580
* 1581
* 1582
* 1583
* 1584
* 1585
* 1586
* 1587
* 1588
* 1589
* 1590
* 1591
* 1592
* 1593
* 1594
* 1595
* 1596
* 1597
* 1598
* 1599
* 1600
* 1601
* 1602
* 1603
* 1604
* 1605
* 1606
* 1607
* 1608
* 1609
* 1610
* 1611
* 1612
* 1613
* 1614
* 1615
* 1616
* 1617
* 1618
* 1619
* 1620
* 1621
* 1622
* 1623
* 1624
* 1625
* 1626
* 1627
* 1628
* 1629
* 1630
* 1631
* 1632
* 1633
* 1634
* 1635
* 1636
* 1637
* 1638
* 1639
* 1640
* 1641
* 1642
* 1643
* 1644
* 1645
* 1646
* 1647
* 1648
* 1649
* 1650
* 1651
* 1652
* 1653
* 1654
* 1655
* 1656
* 1657
* 1658
* 1659
* 1660
* 1661
* 1662
* 1663
* 1664
* 1665
* 1666
* 1667
* 1668
* 1669
* 1670
* 1671
* 1672
* 1673
* 1674
* 1675
* 1676
* 1677
* 1678
* 1679
* 1680
* 1681
* 1682
* 1683
* 1684
* 1685
* 1686
* 1687
* 1688
* 1689
* 1690
* 1691
* 1692
* 1693
* 1694
* 1695
* 1696
* 1697
* 1698
* 1699
* 1700
* 1701
* 1702
* 1703
* 1704
* 1705
* 1706
* 1707
* 1708
* 1709
* 1710
* 1711
* 1712
* 1713
* 1714
* 1715
* 1716
* 1717
* 1718
* 1719
* 1720
* 1721
* 1722
* 1723
* 1724
* 1725
* 1726
* 1727
* 1728
* 1729
* 1730
* 1731
* 1732
* 1733
* 1734
* 1735
* 1736
* 1737
* 1738
* 1739
* 1740
* 1741
* 1742
* 1743
* 1744
* 1745
* 1746
* 1747
* 1748
* 1749
* 1750
* 1751
* 1752
* 1753
* 1754
* 1755
* 1756
* 1757
* 1758
* 1759
* 1760
* 1761
* 1762
* 1763
* 1764
* 1765
* 1766
* 1767
* 1768
* 1769
* 1770
* 1771
* 1772
* 1773
* 1774
* 1775
* 1776
* 1777
* 1778
* 1779
* 1780
* 1781
* 1782
* 1783
* 1784
* 1785
* 1786
* 1787
* 1788
* 1789
* 1790
* 1791
* 1792
* 1793
* 1794
* 1795
* 1796
* 1797
* 1798
* 1799
* 1800
* 1801
* 1802
* 1803
* 1804
* 1805
* 1806
* 1807
* 1808
* 1809
* 1810
* 1811
* 1812
* 1813
* 1814
* 1815
* 1816
* 1817
* 1818
* 1819
* 1820
* 1821
* 1822
* 1823
* 1824
* 1825
* 1826
* 1827
* 1828
* 1829
* 1830
* 1831
* 1832
* 1833
* 1834
* 1835
* 1836
* 1837
* 1838
* 1839
* 1840
* 1841
* 184
```



TRM, INC. LOCATION AND MOVEMENT ANALYSIS SYSTEM  
21 FEB 78 LAMAS FLOW

PAGE 45.001

```
63 * 33 SOLUTION:
64 * 34 PRESENT RESULTS
65 * 35 EXIT:
66 * 36 return TO CALLING PROGRAM
67 * 37 enddo CASE OF
68 * 38 enddo FOREVER
69 * 39 enddo UPON FUNCTION REQUEST
70 *
71 *****
1088 *
1089 *
1090 *
1091 *
1092 *
1093 *
1094 *
```

PERFORM INTERDICTION OPERATIONS

```

REF PAGE
*
* 1 . HERE, THE USER MAY INTERDICTION TO FORCE UNITS TO MOVE A CERTAIN
* 2 . WAY, TO MAKE LINKS IMPASSIBLE, OR TO SEE THE EFFECTS OF SUCH INTERDICTION.
* 3 . THE USER MAY ALSO LOOK AT THE CONTENTS OF NODE+VECTORS.
* 4
* 5 do UPON FUNCTION REQUEST
* 6 do FOREVER
* 7 DISPLAY FUNCTION MENU
* 8 WAIT FOR USER INPUT
* 9 do CASE OF
* 10 VECTOR2:
47 * 11 PRINT MAP+NUMBER, NODE+NUMBER OF NODE+VECTOR NEAREST GIVEN COORDINATES
* 12
* 13 CONTENTS:
48 * 14 CHANGE CONTENTS OF A NODE+VECTOR
* 15
50 * 16 DELETE A NODE FROM THE NODE+VECTOR ARRAY
* 17
51 * 18 PRINT CONTENTS OF A NODE+VECTOR
* 19
52 * 20 CHANGE PROPERTIES OF A UNIT
* 21 return TO CALLING PROGRAM
* 22 enddo CASE OF
* 23 enddo FOREVER
* 24 enddo UPON FUNCTION REQUEST
*
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119

```



TRI, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
LAMAS FLOW

PAGE 47

PRINT MAP+NUMBER, NODE+NUMBER OF NODE+VECTOR NEAREST GIVEN COORDINATES

REF  
PAGE

```
1  . THIS ALLOWS THE USER TO FIND OUT WHICH NODE+VECTOR IS NEAREST ANY UTM
2  .COORDINATES OF INTEREST. THIS IS OFTEN HELPFUL IF THE TRACKBALL IS
3  .UNAVAILABLE.
4  .
5  . PROMPT FOR USER INPUT UTM COORDINATES
6  . SET LEAST TO INFINITY
7  . DO FOR EACH NODE+VECTOR
8  .   DISTANCE = (UTM LAT.-LATITUDE) ** 2 + (UTM LON. - LONGITUDE) **2
9  .   IF DISTANCE IS LESS THAN LEAST
10 .     SET LEAST EQUAL TO DISTANCE
11 .     REMEMBER NODE+VECTOR
12 .   endif DISTANCE IS LESS THAN LEAST
13 . enddo FOR EACH NODE+VECTOR
14 .
15 . THE ANSWER IS THE LAST REMEMBERED NODE+VECTOR
16 .
17 . PRINT OUT MAP+NUMBER, NODE+NUMBER OF NODE+VECTOR
```

1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137

TRU, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
LAMAS FLIN

PAGE 43

# CHANGE CONTENTS OF A NODE+VECTOR

REF  
PAGE

```

1  . THIS FUNCTION ALLOWS THE USER TO CHANGE CERTAIN CONDITIONS OF ANY
2  .NODE+VECTOR. THE USER MAY CHANGE ANY OF SIX LINK CHARACTERISTICS,
3  .NUMBER OF LANES, LINK DISTANCE, TERRAIN CODE, BRIDGE CODE, CITY CODE,
4  .OR ROAD TYPE.
5  .
6  .do UPON FUNCTION REQUEST
7  .PROMPT ENTRY OF MAP+NUMBER, NODE+NUMBER
8  .if ENTERED NODE+VECTOR IS NOT IN MEMORY
9  .PRINT ERROR MESSAGE
10 .TRY AGAIN
11 .else ENTERED NODE+VECTOR IS IN MEMORY
12 .PROMPT FOR LINKING MAP+NUMBER, NODE+NUMBER
13 .if LINKING NODE+VECTOR IS NOT IN MEMORY
14 .PRINT ERROR MESSAGE
15 .TRY AGAIN
16 .else LINKING NODE+VECTOR IS IN MEMORY
17 .if LINKING NODE+VECTOR DOES NOT LINK TO FIRST ENTERED NODE+VECTOR
18 .PRINT ERROR MESSAGE
19 .TRY AGAIN
20 .else TWO NAMED NODE+VECTORS ARE IN MEMORY AND ARE LINKED
21 .PRINT LIST OF CHOICES FOR ALTERATION
22 .WAIT FOR USER INPUT
23 .do CASE OF
24 .ROAD:
25 .CHANGE CHARACTERISTIC ...ROAD TYPE
26 .LINK:
27 .CHANGE CHARACTERISTIC ...LINK DISTANCE
28 .TERRAIN:
29 .CHANGE CHARACTERISTIC ...TERRAIN CODE
30 .LANES:
31 .CHANGE CHARACTERISTIC ...NUMBER IF LANES
32 .BRIDGE:

```

1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170



TRM, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
LAMAS FLTH

PAGE 48.001

```
49 * 33      CHANGE CHARACTERISTIC ...BRIDGE CODE      * 1171
49 * 34 CITY:                                     * 1172
49 * 35      CHANGE CHARACTERISTIC ...CITY CODE        * 1173
49 * 36      enddo CASE OF                             * 1174
49 * 37      return TO CALLING PROGRAM                 * 1175
49 * 38      endif LINKING NODE+VECTOR DOES NOT LINK TO FIRST ENTERED NODE+VECTOR * 1176
49 * 39      endif LINKING NODE+VECTOR IS NOT IN MEMORY * 1177
49 * 40      endif ENTERED NODE+VECTOR IS NOT IN MEMORY * 1178
49 * 41      enddo UPON FUNCTION REQUEST                * 1179
*
*****
```

TRM, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
LAMPAS FLIN

PAGE 45

CHANGE CHARACTERISTIC

REF  
PAGE

```
1 1 . THIS IS THE SAME PROCEDURE REGARDLESS OF WHICH CHARACTERISTIC IS BEING
2 2 .CHANGED.
3 3 .
4 4 PRINT OUT OLD VALUE
5 5 PROMPT FOR NEW VALUE
6 6 IF NEW VALUE WITHIN ACCEPTABLE LIMITS
7 7 ASSIGN NEW VALUE TO NODE+VECTOR
8 8 else NEW VALUE IS NOT ACCEPTABLE
9 9 PRINT ERROR MESSAGE
10 10 TRY AGAIN
11 11 endif NEW VALUE IS WITHIN ACCEPTABLE LIMITS
```

1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191



DELETE A NODE FROM THE NODE+VECTOR ARRAY

REF  
PAGE

```

1  . FOR THE PURPOSE OF INTERDICTION IT MAY BE DESIRED THAT A NODE+VECTOR
2  .BE 'ERASED', THAT IS, AFFECTED SO THAT NO UNIT WILL TRAVEL THROUGH IT.
3  .THIS ROUTINE CHANGES THE NUMBER OF LANES PARAMETER FOR EACH ADJACENT
4  .LINK TO 0. IF THERE'S NO ROAD, NO UNIT CAN TRAVEL THERE. ONCE
5  .DONE, RESTORATION IS PERFORMED BY EITHER REESTABLISHING EACH LINK
6  .MANUALLY, OR BY READING THE MAPS FROM DISK AGAIN.
7
8  .
9  .do UPON FUNCTION REQUEST
10 .PROMPT FOR MAP+NUMBER, NODE+NUMBER OF NODE+VECTOR TO BE ERASED
11 .if ENTERED VALUES ARE INVALID
12 .PRINT ERROR MESSAGE
13 .TRY AGAIN
14 .else ENTERED VALUES ARE VALID
15 .do FOR EACH ADJACENT NODE
16 .SET NUMBER OF LANES TO THIS NODE TO ZERO
17 .enddo FOR EACH ADJACENT NODE
18 .return TO CALLING PROGRAM
19 .endif ENTERED VALUES ARE INVALID
20 .enddo UPON FUNCTION REQUEST

```

1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211

PRINT CONTENTS OF A NODE+VECTOR

REF  
PAGE

```
*****  
* 1 do UPON FUNCTION REQUEST  
* 2   PROMPT FOR MAP+NUMBER, NODE+NUMBER  
* 3   if ENTERED VALUES ARE NOT VALID  
* 4     PRINT ERROR MESSAGE  
* 5     TRY AGAIN  
* 6   else ENTERED VALUES ARE VALID  
* 7     PRINT CONTENTS OF NODE+VECTOR  
* 8     return TO CALLING PROGRAM  
* 9   endif ENTERED VALUES ARE NOT VALID  
* 10 enddo UPON FUNCTION REQUEST  
*****
```

1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222



TRW, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
LAWAS FLOW

PAGE 52

CHANGE PROPERTIES OF A UNIT

REF  
PAGE

```
*****
* 1 . AT THIS TIME, THE ONLY PROPERTY WHICH WE WILL ALLOW TO BE CHANGED,
* 2 . IS THE UNIT'S PRIORITY.
* 3 .
* 4 do UPON FUNCTION REQUEST
* 5   PROMPT ENTRY OF UNIT NAME
* 6   if INPUT NAME HAS AN ENTRY IN UNITS FILE
* 7     PRINT OLD VALUE
* 8   PROMPT ENTRY OF NEW PRIORITY
* 9   if INPUT IS BETWEEN 0 AND 127
* 10     SET NEW VALUE IN UNIT<VECTOR
* 11     return TO CALLING PROGRAM
* 12   else INPUT IS NOT VALID
* 13     PRINT ERROR MESSAGE
* 14     TRY AGAIN
* 15   endif INPUT IS BETWEEN 0 AND 127
* 16   else INPUT NAME HAS NO ENTRY IN UNITS FILE
* 17     PRINT ERROR MESSAGE
* 18     TRY AGAIN
* 19   endif INPUT NAME HAS AN ENTRY IN UNITS FILE
* 20 enddo UPON FUNCTION REQUEST
*****
```

1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243

TRH, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
LAMAS FLOW

PAGE 53

CALCULATE PATH(S) AND SOLUTION VECTORS

REF  
PAGE

```

1  THIS ROUTINE WILL DETERMINE THE BEST PATH FOR GROUND FORCE
2  .UNITS TO USE, TAKING INTO CONSIDERATION TRAVEL TIME, GROUND COVER, TER-
3  .RAIN, BRIDGES, AND CITIES.
4  .BEST DOESN'T ONLY IMPLY FASTEST, BUT MAY MEAN LEAST RISKY, OR SOME
5  .COMBINATION OF SPEED AND RISK.
6  .THERE ARE SEVERAL OPTIONS FROM WHICH THE USER IS FREE TO CHOOSE. AMONG
7  .THESE ARE:
8  . 1) SIMPLE BEST PATH CALCULATION, REGARDLESS OF TIME CONFLICTS,
9  . 2) AUTOMATIC CONFLICT RESOLUTION; THE PROGRAM DOES THE RESOLUTION
10 . WITHOUT USER INTERVENTION.
11 . 3) WORK FORWARDS IN TIME, THAT IS, GIVE A STARTING TIME AND NODE,
12 . AND A DESTINATION NODE. THE PROGRAM WILL THEN WORK FORWARD IN TIME,
13 . BEGINNING AT THE START NODE, AND ENDING AT THE DESTINATION NODE.
14 . 4) WORK BACKWARDS IN TIME. GIVEN A STARTING NODE, DESTINATION NODE
15 . AND ARRIVAL TIME, START AT THE DESTINATION NODE, WORK BACKWARDS IN TIME
16 . TO THE STARTING NODE.
17 . AFTER THE PATH HAS BEEN DETERMINED, THE ROUTE IS RECONSTRUCTED USING
18 . SOLUTION+VECTORS. THESE VECTORS RECORD THE TIMES DURING WHICH THE VAR-
19 . IOUS NODES ARE BUSY. THIS INFORMATION WILL BE USED WHEN RESOLVING
20 . TIME CONFLICTS AMONG THE PATHS.
21 .
22 . DO UPON FUNCTION REQUEST
23 .
24 . BEFORE ANYTHING ELSE, CLEAR AWAY ALL OLD ROUTES AND SOLUTIONS
25 .
26 . PURGE ALL ROUTES
27 .
28 . NUN GET PATH PARAMETERS
29 .
30 . PROMPT FOR ENTRY OF MOVEMENT TYPE (BACKWARD OR FORWARD IN TIME)
31 .
32 . PROMPT FOR ENTRY OF TIME CONFLICT RESOLUTION CODE (NONE OR AUTOMATIC)
33 .
34 .
35 .
36 .
37 .
38 .
39 .
40 .
41 .
42 .
43 .
44 .
45 .
46 .
47 .
48 .
49 .
50 .
51 .
52 .
53 .
54 .
55 .
56 .
57 .
58 .
59 .
60 .
61 .
62 .
63 .
64 .
65 .
66 .
67 .
68 .
69 .
70 .
71 .
72 .
73 .
74 .
75 .
76 .
77 .
78 .
79 .
80 .
81 .
82 .
83 .
84 .
85 .
86 .
87 .
88 .
89 .
90 .
91 .
92 .
93 .
94 .
95 .
96 .
97 .
98 .
99 .
100 .
101 .
102 .
103 .
104 .
105 .
106 .
107 .
108 .
109 .
110 .
111 .
112 .
113 .
114 .
115 .
116 .
117 .
118 .
119 .
120 .
121 .
122 .
123 .
124 .
125 .
126 .
127 .
128 .
129 .
130 .
131 .
132 .
133 .
134 .
135 .
136 .
137 .
138 .
139 .
140 .
141 .
142 .
143 .
144 .
145 .
146 .
147 .
148 .
149 .
150 .
151 .
152 .
153 .
154 .
155 .
156 .
157 .
158 .
159 .
160 .
161 .
162 .
163 .
164 .
165 .
166 .
167 .
168 .
169 .
170 .
171 .
172 .
173 .
174 .
175 .
176 .
177 .
178 .
179 .
180 .
181 .
182 .
183 .
184 .
185 .
186 .
187 .
188 .
189 .
190 .
191 .
192 .
193 .
194 .
195 .
196 .
197 .
198 .
199 .
200 .
201 .
202 .
203 .
204 .
205 .
206 .
207 .
208 .
209 .
210 .
211 .
212 .
213 .
214 .
215 .
216 .
217 .
218 .
219 .
220 .
221 .
222 .
223 .
224 .
225 .
226 .
227 .
228 .
229 .
230 .
231 .
232 .
233 .
234 .
235 .
236 .
237 .
238 .
239 .
240 .
241 .
242 .
243 .
244 .
245 .
246 .
247 .
248 .
249 .
250 .
251 .
252 .
253 .
254 .
255 .
256 .
257 .
258 .
259 .
260 .
261 .
262 .
263 .
264 .
265 .
266 .
267 .
268 .
269 .
270 .
271 .
272 .
273 .
274 .
275 .
276 .
277 .
278 .
279 .
280 .
281 .
282 .
283 .
284 .
285 .
286 .
287 .
288 .
289 .
290 .
291 .
292 .
293 .
294 .
295 .
296 .
297 .
298 .
299 .
300 .
301 .
302 .
303 .
304 .
305 .
306 .
307 .
308 .
309 .
310 .
311 .
312 .
313 .
314 .
315 .
316 .
317 .
318 .
319 .
320 .
321 .
322 .
323 .
324 .
325 .
326 .
327 .
328 .
329 .
330 .
331 .
332 .
333 .
334 .
335 .
336 .
337 .
338 .
339 .
340 .
341 .
342 .
343 .
344 .
345 .
346 .
347 .
348 .
349 .
350 .
351 .
352 .
353 .
354 .
355 .
356 .
357 .
358 .
359 .
360 .
361 .
362 .
363 .
364 .
365 .
366 .
367 .
368 .
369 .
370 .
371 .
372 .
373 .
374 .
375 .
376 .
377 .
378 .
379 .
380 .
381 .
382 .
383 .
384 .
385 .
386 .
387 .
388 .
389 .
390 .
391 .
392 .
393 .
394 .
395 .
396 .
397 .
398 .
399 .
400 .
401 .
402 .
403 .
404 .
405 .
406 .
407 .
408 .
409 .
410 .
411 .
412 .
413 .
414 .
415 .
416 .
417 .
418 .
419 .
420 .
421 .
422 .
423 .
424 .
425 .
426 .
427 .
428 .
429 .
430 .
431 .
432 .
433 .
434 .
435 .
436 .
437 .
438 .
439 .
440 .
441 .
442 .
443 .
444 .
445 .
446 .
447 .
448 .
449 .
450 .
451 .
452 .
453 .
454 .
455 .
456 .
457 .
458 .
459 .
460 .
461 .
462 .
463 .
464 .
465 .
466 .
467 .
468 .
469 .
470 .
471 .
472 .
473 .
474 .
475 .
476 .
477 .
478 .
479 .
480 .
481 .
482 .
483 .
484 .
485 .
486 .
487 .
488 .
489 .
490 .
491 .
492 .
493 .
494 .
495 .
496 .
497 .
498 .
499 .
500 .
501 .
502 .
503 .
504 .
505 .
506 .
507 .
508 .
509 .
510 .
511 .
512 .
513 .
514 .
515 .
516 .
517 .
518 .
519 .
520 .
521 .
522 .
523 .
524 .
525 .
526 .
527 .
528 .
529 .
530 .
531 .
532 .
533 .
534 .
535 .
536 .
537 .
538 .
539 .
540 .
541 .
542 .
543 .
544 .
545 .
546 .
547 .
548 .
549 .
550 .
551 .
552 .
553 .
554 .
555 .
556 .
557 .
558 .
559 .
560 .
561 .
562 .
563 .
564 .
565 .
566 .
567 .
568 .
569 .
570 .
571 .
572 .
573 .
574 .
575 .
576 .
577 .
578 .
579 .
580 .
581 .
582 .
583 .
584 .
585 .
586 .
587 .
588 .
589 .
590 .
591 .
592 .
593 .
594 .
595 .
596 .
597 .
598 .
599 .
600 .
601 .
602 .
603 .
604 .
605 .
606 .
607 .
608 .
609 .
610 .
611 .
612 .
613 .
614 .
615 .
616 .
617 .
618 .
619 .
620 .
621 .
622 .
623 .
624 .
625 .
626 .
627 .
628 .
629 .
630 .
631 .
632 .
633 .
634 .
635 .
636 .
637 .
638 .
639 .
640 .
641 .
642 .
643 .
644 .
645 .
646 .
647 .
648 .
649 .
650 .
651 .
652 .
653 .
654 .
655 .
656 .
657 .
658 .
659 .
660 .
661 .
662 .
663 .
664 .
665 .
666 .
667 .
668 .
669 .
670 .
671 .
672 .
673 .
674 .
675 .
676 .
677 .
678 .
679 .
680 .
681 .
682 .
683 .
684 .
685 .
686 .
687 .
688 .
689 .
690 .
691 .
692 .
693 .
694 .
695 .
696 .
697 .
698 .
699 .
700 .
701 .
702 .
703 .
704 .
705 .
706 .
707 .
708 .
709 .
710 .
711 .
712 .
713 .
714 .
715 .
716 .
717 .
718 .
719 .
720 .
721 .
722 .
723 .
724 .
725 .
726 .
727 .
728 .
729 .
730 .
731 .
732 .
733 .
734 .
735 .
736 .
737 .
738 .
739 .
740 .
741 .
742 .
743 .
744 .
745 .
746 .
747 .
748 .
749 .
750 .
751 .
752 .
753 .
754 .
755 .
756 .
757 .
758 .
759 .
760 .
761 .
762 .
763 .
764 .
765 .
766 .
767 .
768 .
769 .
770 .
771 .
772 .
773 .
774 .
775 .
776 .
777 .
778 .
779 .
780 .
781 .
782 .
783 .
784 .
785 .
786 .
787 .
788 .
789 .
790 .
791 .
792 .
793 .
794 .
795 .
796 .
797 .
798 .
799 .
800 .
801 .
802 .
803 .
804 .
805 .
806 .
807 .
808 .
809 .
810 .
811 .
812 .
813 .
814 .
815 .
816 .
817 .
818 .
819 .
820 .
821 .
822 .
823 .
824 .
825 .
826 .
827 .
828 .
829 .
830 .
831 .
832 .
833 .
834 .
835 .
836 .
837 .
838 .
839 .
840 .
841 .
842 .
843 .
844 .
845 .
846 .
847 .
848 .
849 .
850 .
851 .
852 .
853 .
854 .
855 .
856 .
857 .
858 .
859 .
860 .
861 .
862 .
863 .
864 .
865 .
866 .
867 .
868 .
869 .
870 .
871 .
872 .
873 .
874 .
875 .
876 .
877 .
878 .
879 .
880 .
881 .
882 .
883 .
884 .
885 .
886 .
887 .
888 .
889 .
890 .
891 .
892 .
893 .
894 .
895 .
896 .
897 .
898 .
899 .
900 .
901 .
902 .
903 .
904 .
905 .
906 .
907 .
908 .
909 .
910 .
911 .
912 .
913 .
914 .
915 .
916 .
917 .
918 .
919 .
920 .
921 .
922 .
923 .
924 .
925 .
926 .
927 .
928 .
929 .
930 .
931 .
932 .
933 .
934 .
935 .
936 .
937 .
938 .
939 .
940 .
941 .
942 .
943 .
944 .
945 .
946 .
947 .
948 .
949 .
950 .
951 .
952 .
953 .
954 .
955 .
956 .
957 .
958 .
959 .
960 .
961 .
962 .
963 .
964 .
965 .
966 .
967 .
968 .
969 .
970 .
971 .
972 .
973 .
974 .
975 .
976 .
977 .
978 .
979 .
980 .
981 .
982 .
983 .
984 .
985 .
986 .
987 .
988 .
989 .
990 .
991 .
992 .
993 .
994 .
995 .
996 .
997 .
998 .
999 .
1000 .
1001 .
1002 .
1003 .
1004 .
1005 .
1006 .
1007 .
1008 .
1009 .
1010 .
1011 .
1012 .
1013 .
1014 .
1015 .
1016 .
1017 .
1018 .
1019 .
1020 .
1021 .
1022 .
1023 .
1024 .
1025 .
1026 .
1027 .
1028 .
1029 .
1030 .
1031 .
1032 .
1033 .
1034 .
1035 .
1036 .
1037 .
1038 .
1039 .
1040 .
1041 .
1042 .
1043 .
1044 .
1045 .
1046 .
1047 .
1048 .
1049 .
1050 .
1051 .
1052 .
1053 .
1054 .
1055 .
1056 .
1057 .
1058 .
1059 .
1060 .
1061 .
1062 .
1063 .
1064 .
1065 .
1066 .
1067 .
1068 .
1069 .
1070 .
1071 .
1072 .
1073 .
1074 .
1075 .
1076 .
1077 .
1078 .
1079 .
1080 .
1081 .
1082 .
1083 .
1084 .
1085 .
1086 .
1087 .
1088 .
1089 .
1090 .
1091 .
1092 .
1093 .
1094 .
1095 .
1096 .
1097 .
1098 .
1099 .
1100 .
1101 .
1102 .
1103 .
1104 .
1105 .
1106 .
1107 .
1108 .
1109 .
1110 .
1111 .
1112 .
1113 .
1114 .
1115 .
1116 .
1117 .
1118 .
1119 .
1120 .
1121 .
1122 .
1123 .
1124 .
1125 .
1126 .
1127 .
1128 .
1129 .
1130 .
1131 .
1132 .
1133 .
1134 .
1135 .
1136 .
1137 .
1138 .
1139 .
1140 .
1141 .
1142 .
1143 .
1144 .
1145 .
1146 .
1147 .
1148 .
1149 .
1150 .
1151 .
1152 .
1153 .
1154 .
1155 .
1156 .
1157 .
1158 .
1159 .
1160 .
1161 .
1162 .
1163 .
1164 .
1165 .
1166 .
1167 .
1168 .
1169 .
1170 .
1171 .
1172 .
1173 .
1174 .
1175 .
1176 .
1177 .
1178 .
1179 .
1180 .
1181 .
1182 .
1183 .
1184 .
1185 .
1186 .
1187 .
1188 .
1189 .
1190 .
1191 .
1192 .
1193 .
1194 .
1195 .
1196 .
1197 .
1198 .
1199 .
1200 .
1201 .
1202 .
1203 .
1204 .
1205 .
1206 .
1207 .
1208 .
1209 .
1210 .
1211 .
1212 .
1213 .
1214 .
1215 .
1216 .
1217 .
1218 .
1219 .
1220 .
1221 .
1222 .
1223 .
1224 .
1225 .
1226 .
1227 .
1228 .
1229 .
1230 .
1231 .
1232 .
1233 .
1234 .
1235 .
1236 .
1237 .
1238 .
1239 .
1240 .
1241 .
1242 .
1243 .
1244 .
1245 .
1246 .
1247 .
1248 .
1249 .
1250 .
1251 .
1252 .
1253 .
1254 .
1255 .
1256 .
1257 .
1258 .
1259 .
1260 .
1261 .
1262 .
1263 .
1264 .
1265 .
1266 .
1267 .
1268 .
1269 .
1270 .
1271 .
1272 .
1273 .
1274 .
1275 .
1276 .
1277 .
1278 .
1279 .
1280 .
1281 .
1282 .
1283 .
1284 .
1285 .
1286 .
1287 .
1288 .
1289 .
1290 .
1291 .
1292 .
1293 .
1294 .
1295 .
1296 .
1297 .
1298 .
1299 .
1300 .
1301 .
1302 .
1303 .
1304 .
1305 .
1306 .
1307 .
1308 .
1309 .
1310 .
1311 .
1312 .
1313 .
1314 .
1315 .
1316 .
1317 .
1318 .
1319 .
1320 .
1321 .
1322 .
1323 .
1324 .
1325 .
1326 .
1327 .
1328 .
1329 .
1330 .
1331 .
1332 .
1333 .
1334 .
1335 .
1336 .
1337 .
1338 .
1339 .
1340 .
1341 .
1342 .
1343 .
1344 .
1345 .
1346 .
1347 .
1348 .
1349 .
1350 .
1351 .
1352 .
1353 .
1354 .
1355 .
1356 .
1357 .
1358 .
1359 .
1360 .
1361 .
1362 .
1363 .
1364 .
1365 .
1366 .
1367 .
1368 .
1369 .
1370 .
1371 .
1372 .
1373 .
1374 .
1375 .
1376 .
1377 .
1378 .
1379 .
1380 .
1381 .
1382 .
1383 .
1384 .
1385 .
1386 .
1387 .
1388 .
1389 .
1390 .
1391 .
1392 .
1393 .
1394 .
1395 .
1396 .
1397 .
1398 .
1399 .
1400 .
1401 .
1402 .
1403 .
1404 .
1405 .
1406 .
1407 .
1408 .
1409 .
1410 .
1411 .
1412 .
1413 .
1414 .
1415 .
1416 .
1417 .
1418 .
1419 .
1420 .
1421 .
1422 .
1423 .
1424 .
1425 .
1426 .
1427 .
1428 .
1429 .
1430 .
1431 .
1432 .
1433 .
1434 .
1435 .
1436 .
1437 .
1438 .
1439 .
1440 .
1441 .
1442 .
1443 .
1444 .
1445 .
1446 .
1447 .
1448 .
1449 .
1450 .
1451 .
1452 .
1453 .
1454 .
1455 .
1456 .
1457 .
1458 .
1459 .
1460 .
1461 .
1462 .
1463 .
1464 .
1465 .
1466 .
1467 .
1468 .
1469 .
1470 .
1471 .
1472 .
1473 .
1474 .
1475 .
1476 .
1477 .
1478 .
1479 .
1480 .
1481 .
1482 .
1483 .
1484 .
1485 .
1486 .
1487 .
1488 .
1489 .
1490 .
1491 .
1492 .
1493 .
1494 .
1495 .
1496 .
1497 .
1498 .
1499 .
1500 .
1501 .
1502 .
1503 .
1504 .
1505 .
1506 .
1507 .
1508 .
1509 .
1510 .
1511 .
1512 .
1513 .
1514 .
1515 .
1516 .
1517 .
1518 .
1519 .
1520 .
1521 .
1522 .
1523 .
1524 .
1525 .
1526 .
1527 .
1528 .
1529 .
1530 .
1531 .
1532 .
1533 .
1534 .
1535 .
1536 .
1537 .
1538 .
1539 .
1540 .
1541 .
1542 .
1543 .
1544 .
1545 .
1546 .
1547 .
1548 .
1549 .
1550 .
1551 .
1552 .
1553 .
1554 .
1555 .
1556 .
1557 .
1558 .
1559 .
1560 .
1561 .
1562 .
1563 .
1564 .
1565 .
1566 .
1567 .
1568 .
1569 .
1570 .
1571 .
1572 .
1573 .
1574 .
1575 .
1576 .
1577 .
1578 .
1579 .
1580 .
1581 .
1582 .
1583 .
1584 .
1585 .
1586 .
1587 .
1588 .
1589 .
1590 .
1591 .
1592 .
1593 .
1594 .
1595 .
1596 .
1597 .
1598 .
1599 .
1600 .
1601 .
1602 .
1603 .
1604 .
1605 .
1606 .
1607 .
1608 .
1609 .
1610 .
1611 .
1612 .
1613 .
1614 .
1615 .
1616 .
1617 .
1618 .
1619 .
1620 .
1621 .
1622 .
1623 .
1624 .
1625 .
1626 .
1627 .
1628 .
1629 .
1630 .
1631 .
1632 .
1633 .
1634 .
1635 .
1636 .
1637 .
1638 .
1639 .
1640 .
1641 .
1642 .
1643 .
1644 .
1645 .
1646 .
1647 .
1648 .
1649 .
1650 .
1651 .
1652 .
1653 .
1654 .
1655 .
1656 .
1657 .
1658 .
1659 .
1660 .
1661 .
1662 .
1663 .
1664 .
1665 .
1666 .
1667 .
1668 .
1669 .
1670 .
1671 .
1672 .
1673 .
1674 .
1675 .
1676 .
1677 .
1678 .
1679 .
1680 .
1681 .
1682 .
1683 .
1684 .
1685 .
1686 .
1687 .
1688 .
1689 .
1690 .
1691 .
1692 .
1693 .
1694 .
1695 .
1696 .
1697 .
1698 .
1699 .
1700 .
1701 .
1702 .
1703 .
1704 .
1705 .
1706 .
1707 .
1708 .
1709 .
1710 .
1711 .
1712 .
1713 .
1714 .
1715 .
1716 .
1717 .
1718 .
1719 .
1720 .
1721 .
1722 .
1723 .
1724 .
1725 .
1726 .
1727 .
1728 .
1729 .
1730 .
1731 .
1732 .
1733 .
1734 .
1735 .
1736 .
1737 .
1738 .
1739 .
1740 .
1741 .
1742 .
1743 .
1744 .
1745 .
1746 .
1747 .
1748 .
1749 .
1750 .
1751 .
1752 .
1753 .
1754 .
1755 .
1756 .
1757 .
1758 .
1759 .
1760 .
1761 .
1762 .
1763 .
1764 .
1765 .
1766 .
1767 .
1768 .
1769 .
1770 .
1771 .
1772 .
1773 .
1774 .
1775 .
1776 .
1777 .
1778 .
1779 .
1780 .
1781 .
1782 .
1783 .
1784 .
1785 .
1786 .
1787 .
1788 .
1789 .
1790 .
1791 .
1792 .
1793 .
1794 .
1795 .
1796 .
1797 .
1798 .
1799 .
1800 .
1801 .
1802 .
1803 .
1804 .
1805 .
1806 .
1807 .
1808 .
1809 .
1810 .
1811 .
1812 .
1813 .
1814 .
1815 .
1816 .
1817 .
1818 .
1819 .
1820 .
1821 .
1822 .
1823 .
1824 .
1825 .
1826 .
1827 .
1828 .
1829 .
1830 .
1831 .
1832 .
1833 .
1834 .
1835 .
1836 .
1837 .
1838 .
1839 .
1840 .
1841 .
1842 .
1843 .
1844 .
1845 .
1846 .
1847 .
1848 .
1849 .
1850 .
1851 .
1852 .
1853 .
1854 .
1855 .
1856 .
1857 .
1858 .
1859 .
1860 .
1861 .
1862 .
1863 .
1864 .
1865 .
1866 .
1867 .
1868 .
1869 .
1870 .
1871 .
1872 .
1873 .
1874 .
1875 .
1876 .
1877 .
1878 .
1879 .
1880 .
1881 .
1882 .
1883 .
1884 .
1885 .
1886 .
1887 .
1888 .
1889 .
1890 .
1891 .
1892 .
1893 .
1894 .
1895 .
1896 .
1897 .
1898 .
1899 .
1900 .
1901 .
1902 .
1903 .
1904 .
1905 .
1906 .
1907 .
1908 .
1909 .
1910 .
1911 .
1912 .
1913 .
1914 .
1915 .
1916 .
1917 .
1918 .
1919 .
1920 .
1921 .
1922 .
1923 .
1924 .
1925 .
1926 .
1927 .
1928 .
1929 .
1930 .
1931 .
1932 .
1933 .
1934 .
1935 .
1936 .
1937 .
1938 .
1939 .
1940 .
1941 .
1942 .
1943 .
1944 .
1945 .
1946 .
1947 .
1948 .
1949 .
1950 .
1951 .
1952 .
1953 .
1954 .
1955 .
1956 .
1957 .
1958 .
1959 .
1960 .
1961 .
1962 .
1963 .
1964 .
1965 .
1966 .
1967 .
1968 .
1969 .
1970 .
1971 .
1972 .
1973 .
1974 .
1975 .
1976 .
1977 .
1978 .
1979 .
1980 .
1981 .
1982 .
1983 .
1984 .
1985 .
1986 .
1987 .
1988 .
1989 .
1990 .
1991 .
1992 .
1993 .
1994 .
1995 .
1996 .
1997 .
1998 .
1999 .
2000 .
2001 .
2002 .
2003 .
2004 .
2005 .
2006 .
2007 .
2008 .
2009 .
2010 .
2011 .
2012 .
2013 .
2014 .
2015 .
2016 .
2017 .
2018 .
2019 .
2020 .
2021 .
2022 .
2023 .
2024 .
2025 .
2026 .
2027 .
2028 .
2029 .
2030 .
2031 .
2032 .
2033 .
2034 .
2035 .
2036 .
2037 .
2038 .
2039 .
2040 .
2041 .
2042 .
2043 .
2044 .
2045 .
2046 .
2047 .
2048 .
2049 .
2050 .
2051 .
2052 .
2053 .
2054 .
2055 .
2056 .
2057 .
2058 .
2059 .
2060 .
2061 .
2062 .
2063 .
2064 .
2065 .
2066 .
2067 .
2068 .
2069 .
2070 .
2071 .
2072 .
2073 .
2074 .
2075 .
2076 .
2077 .
2078 .
2079 .
2080 .
2081 .
2082 .
2083 .
2084 .
2085 .
2086 .
2087 .
2088 .
2089 .
2090 .
2091 .
2092 .
2093 .
2094 .
2095 .
2096 .
2097 .
2098 .
2099 .
2100 .
2101 .
2102 .
2103 .
2104 .
2105 .
2106 .
2107 .
2108 .
2109 .
2110 .
2111 .
2112 .
2113 .
2114 .
2115 .
2116 .
2117 .
2118 .
2119 .
2120 .
2121 .
2122 .
2123 .
2124 .
2125 .
2126 .
2127 .
2128 .
2129 .
2130 .
2131 .
2132 .
2133 .
2134 .
2135 .
2136 .
2137 .
2138 .
2139 .
2140 .
2141 .
2142 .
2143 .
2144 .
2145 .
2146 .
2
```





```

* 71 * 1315
* 72 * 1316
* 73 * 1317
* 74 * 1318
* 75 * 1319
* 76 * 1320
* 77 * 1321
55 * 78 * 1322
* 79 * 1323
* 80 * 1324
* 81 * 1325
* 82 * 1326
55 * 83 * 1327
* 84 * 1328
* 85 * 1329
* 86 * 1330
* 87 * 1331
* 88 * 1332
* 89 * 1333
* 90 * 1334
* 91 * 1335
* 92 * 1336
* 93 * 1337
* 94 * 1338
* 95 * 1339
* 96 * 1340
* 97 * 1341
* 98 * 1342
* 99 * 1343
* 00 * 1344
* 01 * 1345
* 02 * 1346
* 03 * 1347
* 04 * 1348
* 05 * 1349
* 06 * 1350
* 07 * 1351
56 * 08 * 1352

.QUANTITIES NECESSARY FOR THE PATH ALGORITHM. THESE ARE NORTH MEASURE,
.PREDECESSOR NODE NUMBER, CUMULATIVE TIME, CUMULATIVE NORTH MEASURE,
.TIME MEASURE, SOLUTION+VECTOR POINTER, LANE USED, RISK MEASURE, AND
.PARK TIME. THESE VALUES ARE HELD IN THE NODE+VECTOR FOR EASY REFER-
.ENCE.
.
if TIME CONFLICT RESOLUTION IS AUTOMATIC
  REINITIALIZE 'WORKING LIST' NODE+VECTOR ENTRIES
  SET K TO FIRST NODE TO BE CONSIDERED (DEPENDENT ON MOVEMENT TYPE)
else TIME CONFLICT RESOLUTION IS NOT AUTOMATIC
  if THIS UNIT'S MOVEMENT TYPE, ENTERED TIME, AND UNIT TYPE DIFFERS FROM/
  THE PREVIOUS UNIT'S
    REINITIALIZE 'WORKING LIST' NODE+VECTOR ENTRIES
    SET K TO FIRST NODE TO BE CONSIDERED (DEPENDENT ON MOVEMENT TYPE)
  endif THIS UNIT'S PARAMETERS DIFFER
endif TIME CONFLICT RESOLUTION IS AUTOMATIC
.
.DURING THIS ALGORITHM, K IS THE NODE FROM WHICH THE ALGORITHM STARTS.
.THIS IS DETERMINED BY THE MOVEMENT TYPE TO BE EITHER THE START NODE
.(TYPE IS FORWARD) OR THE DESTINATION NODE (TYPE IS BACKWARD). I IS THE
.FINAL NODE, THE OPPOSITE OF K (IF K IS THE START NODE, I IS THE DESTI-
.NATION, AND VICE VERSA). J IS THE NODE UNDER CURRENT CONSIDERATION BY
.THE ALGORITHM.
.
SET I TO END NODE (DEPENDENT UPON MOVEMENT TYPE)
SET J TO K
do UNTIL NODE I IS LABELED ... TEST IMMEDIATELY FOR EARLY EXIT
  if I IS LABELED
    undo UNTIL NODE I IS LABELED
  endif I IS LABELED
  if J ISN'T LABELED
    LABEL NODE J
  SET J'S CUMULATIVE TIME
  do FOR EACH OF J'S ADJACENT NODES
    if IT IS LABELED
      cycle FOR EACH OF J'S ADJACENT NODES
    else IT IS NOT LABELED
      CALCULATE AND ASSIGN TIME AND NORTH VALUES
    endif
  enddo
enddo

```



TRK, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
LAMAS FLOW

PAGE 53.003

```
* 09      endif IT IS LABELED
* 10      enddo FOR EACH OF J'S ADJACENT NODES
* 11      endif J ISN'T LABELED
57 * 12      COMPUTE NEXT NODE TO LABEL ...DEPENDENT UPON TYPE
* 13      SET J TO NEW NODE
* 14      enddo UNTIL NODE I LABELED
59 * 15      COMPUTE SOLUTION+VECTORS AND ROUTE+VECTOR
* 16      enddo FOR EACH UNIT
* 17      return TO CALLING PROGRAM
* 18      enddo UPON FUNCTION REQUEST
*
*****
***** TOO MANY LINES IN SEGMENT *****
*
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
```

PURGE ALL ROUTES

REF  
PAGE

```

1  . THIS ROUTINE WILL GET RID OF ALL PREVIOUS PATHS WHICH HAVE BEEN
2  . CALCULATED. THIS CONSISTS OF ERASING ALL SOLUTION+VECTORS AND
3  . ROUTE+VECTORS, RESETTING THEIR COUNTERS TO ZERO, AND ERASING ALL POINT-
4  . ERS WHICH REFERENCE THESE ARRAYS.
5  .
6  . DO FOR EACH ROUTE+VECTOR
7  .   SET CONTENTS TO 0
8  . ENDDO FOR EACH ROUTE+VECTOR
9  . DO FOR EACH SOLUTION+VECTOR
10 .   SET CONTENTS TO 0
11 . ENDDO FOR EACH SOLUTION+VECTOR
12 . SET COUNTERS TO 0
13 . DO FOR EACH NODE+VECTOR
14 .   SET SOLUTION+VECTOR POINTER TO 0
15 . ENDDO FOR EACH NODE+VECTOR

```

1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378



REINITIALIZE 'WORKING LIST' NODE+VECTOR ENTRIES

REF  
PAGE

```

1  THIS FUNCTION MUST BE PERFORMED EACH TIME WE WISH TO
2  "WIPE THE SLATE CLEAN" AND START A NEW PATH-FINDING PROCEDURE.
3
4  do FOR EACH NODE+VECTOR
5      SET TIME MEASURE TO ZERO
6      SET NORTH MEASURE TO ZERO
7      SET CUMULATIVE TIME TO ZERO
8      SET CUMULATIVE NORTH TO 32767
9      SET LABEL TO ZERO
10     SET PREDECESSOR TO -1
11     SET LANE TO ONE
12     SET PARKING TIME TO ZERO
13     SET RISK MEASURE TO 0
14     enddo FOR EACH NODE+VECTOR

```

1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393

CALCULATE AND ASSIGN TIME AND NORTH VALUES

REF  
PAGE

```

1  * WHEN A NODE IS FOUND BY THE PATH ALGORITHM TO BE THE NEXT TO BE
2  * LABELED, THE TIME AND NORTH VALUES OF ALL OF ITS LINKS TO UNLABELED
3  * ADJACENT NODES ARE CALCULATED AND ASSIGNED. THAT PROCESS OF CALCULAT-
4  * ING AND ASSIGNING IS DONE BY THIS ROUTINE.
5  * THERE ARE TWO TYPES OF TIME AND NORTH: MEASURE AND CUMULATIVE. TIME
6  * MEASURE AND NORTH MEASURE REFER TO THE VALUE CALCULATED FOR THE LINK
7  * BETWEEN A NODE AND ONE OF ITS UNLABELED ADJACENT NODES. CUMULATIVE
8  * TIME AND CUMULATIVE NORTH REFER TO THE ACCUMULATED TIME AND NORTH
9  * WHICH HAS BUILT UP DURING THE PATH CALCULATION. THUS CUMULATIVE TIME
10 * IS THE TIME OF DAY AT WHICH A UNIT ARRIVES AT THE ADJACENT NODE, AND
11 * CUMULATIVE NORTH IS THE NORTH OF THE PATH AT THAT POINT IN TIME.
12 * NORTH IS A COMBINATION OF RISK AND TIME, USING PARAMETERS SPECIFIED
13 * BY THE USER. THE EQUATION IS  $C * RISK + K * TIME$ , WHERE C MAY BE
14 * 0 OR 1, AND K MAY BE BETWEEN 0 AND 9.99, BUT BOTH MAY NOT EQUAL 0
15 * AT THE SAME TIME. A BEST PATH IS A PATH WHICH MINIMIZES NORTH. BY
16 * ADJUSTING THE PARAMETERS C AND K, THE USER MAKES HIS OWN DEFINITION
17 * OF BEST. IT COULD BE THAT BEST IS THE LEAST RISKY (C=1, K=0), FASTEST
18 * (C=0, K=1), OR A COMBINATION OF THE TWO.
19 * RISK IS CALCULATED BY SUMMING THE NUMERICAL VALUES ASSIGNED TO THE
20 * FOUR RISK FACTORS: GROUND COVER, TERRAIN, BRIDGE, AND CITY CODES.
21 *
22 * CALCULATE LINK TRAVERSAL TIME
23 * IF MOVEMENT TYPE IS FORWARD
24 *   CUMULATIVE TIME IS PREDECESSOR'S CUMULATIVE TIME + LINK TIME
25 * else MOVEMENT TYPE IS BACKWARD
26 *   CUMULATIVE TIME IS THIS NODE'S CUMULATIVE TIME - LINK TIME
27 * endif MOVEMENT TYPE IS FORWARD
28 *
29 * .RESOLVE TIME CONFLICTS
30 *
31 * IF USER DOES NOT WANT TIME CONFLICTS RESOLVED
32 *   CONTINUE

```

1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426



TRW, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
LANAS FLOW

PAGE 56.001

```
* 33      elseif THIS NODE+VECTOR HAS NEVER BEEN USED BY A PATH
* 34      CONTINUE
* 35      else THIS NODE+VECTOR HAS BEEN USED BY A PATH
* 36
* 37      .LOOK FOR TIME CONFLICT. IF ONE IS FOUND, RESOLVE IT
* 38
* 39      SET ALL FOUR LANES TO CUMULATIVE TIME
* 40      do FOR EACH SOLUTION+VECTOR REFERING TO THIS NODE
* 41      if THE TIME INTERVAL SHOULD NOT BE CONSIDERED
* 42      cycle FOR EACH SOLUTION+VECTOR
* 43      else THE TIME INTERVAL SHOULD BE CONSIDERED
* 44      SET INDEX TO SOLUTION+VECTOR 'S LANE NUMBER
* 45      if THIS UNIT CAN CLEAR THIS NODE BEFORE THE TIME INTERVAL
* 46      cycle FOR EACH SOLUTION+VECTOR
* 47      else THIS UNIT CANNOT CLEAR THIS NODE BEFORE THE TIME INTERVAL
* 48      if THIS UNIT DOES NOT ARRIVE UNTIL AFTER THE TIME INTERVAL
* 49      cycle FOR EACH SOLUTION+VECTOR
* 50      else THIS UNIT ARRIVES DURING THE TIME INTERVAL
* 51
* 52      .THERE IS A TIME CONFLICT
* 53
* 54      if MOVEMENT TYPE IS FORWARD
* 55      HOLD UP UNIT UNTIL NODE CLEARS
* 56      RECORD THIS TIME IN LANE (INDEX)
* 57      else MOVEMENT TYPE IS BACKWARD
* 58      HAVE UNIT ARRIVE BEFORE TIME INTERVAL
* 59      RECORD THIS TIME IN LANE (INDEX)
* 60      endif MOVEMENT TYPE IS FORWARD
* 61
* 62      .THAT'S ALL FOR THIS SOLUTION+VECTOR
* 63
* 64      endif THIS UNIT DOES NOT ARRIVE UNTIL AFTER THE TIME INTERVAL
* 65      endif THIS UNIT CAN CLEAR THIS NODE BEFORE THE TIME INTERVAL
* 66      endif THE TIME INTERVAL SHOULD NOT BE CONSIDERED
* 67      enddo FOR EACH SOLUTION+VECTOR REFERING TO THIS NODE
* 68
* 69      .FIND THE BEST LANE (FOR FORWARD, BEST IS LEAST TIME, FOR BACKWARD, BEST
* 70      .IS THE GREATEST TIME).
```

```

* 71 * if MOVEMENT IS FORWARD
* 72 *   SET BEST TO 32767
* 73 * else MOVEMENT IS BACKWARD
* 74 *   SET BEST TO 0
* 75 * endif MOVEMENT IS FORWARD
* 76 * do FOR EACH AVAILABLE LANE FOR THIS LINK
* 77 *   if TIME OF LANE IS LESS (GREATER) THAN BEST
* 78 *     SET BEST TO TIME OF LANE
* 79 *     REMEMBER LANE NUMBER
* 80 *   endif TIME OF LANE IS LESS (GREATER) THAN BEST
* 81 *   enddo FOR EACH AVAILABLE LANE FOR THIS LINK
* 82 *
* 83 * .ANY TIME CONFLICT HAS NOW BEEN RESOLVED. THE BEST RESULT FROM THE
* 84 * .PREVIOUS SECTION IS NOW THE 'TIME OF RECORD'.
* 85 *
* 86 * .CALCULATE PARKING TIME ('TIME OF RECORD' LESS CUMULATIVE TIME)
* 87 *
* 88 * .PARKING TIME AFFECTS THE ENTIRE LENGTH OF THE UNIT, SO IT MUST BE
* 89 * .SEEN IF THE ADDITION OF THIS PARKING TIME CREATES ANY NEW CONFLICTS.
* 90 * .IF SO, THIS LINK CANNOT BE USED.
* 91 *
* 92 * if PARKING TIME EQUALS ZERO
* 93 *   CONTINUE
* 94 * else PARKING TIME DOES NOT EQUAL ZERO
* 95 *   if PARKING TIME CREATES NEW CONFLICTS
* 96 *     MAKE NO ASSIGNMENTS OF TIME AND WORTH
* 97 *     return
* 98 *   endif PARKING TIME CREATES NEW CONFLICTS
* 99 *   endif PARKING TIME EQUALS ZERO
* 00 * endif THE USER DOES NOT WANT TIME CONFLICTS RESOLVED
* 01 *
* 02 * .NOW CALCULATE RISK AND MAKE ASSIGNMENTS
* 03 *
* 04 * .CALCULATE RISK
* 05 *
* 06 * .CALCULATE WORTH FOR THIS LINK
* 07 * .CALCULATE CUMULATIVE WORTH
* 08 * if CUMULATIVE WORTH IS LESS THAN CURRENT CUMULATIVE WORTH

```

```

* 1465 *
* 1466 *
* 1467 *
* 1468 *
* 1469 *
* 1470 *
* 1471 *
* 1472 *
* 1473 *
* 1474 *
* 1475 *
* 1476 *
* 1477 *
* 1478 *
* 1479 *
* 1480 *
* 1481 *
* 1482 *
* 1483 *
* 1484 *
* 1485 *
* 1486 *
* 1487 *
* 1488 *
* 1489 *
* 1490 *
* 1491 *
* 1492 *
* 1493 *
* 1494 *
* 1495 *
* 1496 *
* 1497 *
* 1498 *
* 1499 *
* 1500 *
* 1501 *
* 1502 *

```



TRU, INC. LOCATION AND MOVEMENT ANALYSIS SYSTEM  
21 FEB 78 LONAS FLIN

1503  
1504

\*  
\*  
\*

09 MAKE ASSIGNMENTS OF TIME, NORTH, PARKING, AND LANE  
10 endif CUMULATIVE NORTH IS LESS THAN CURRENT CUMULATIVE NORTH  
\*

\*\*\*\*\*

\*\*\*\*\* YOU MANY LINES IN SEGMENT

COMPUTE NEXT NODE TO LABEL

REF  
PAGE

```

* 1 *
* 2 *
* 3 *
* 4 *
* 5 *
* 6 *
* 7 *
* 8 *
* 9 *
* 10 *
* 11 *
* 12 *
* 13 *
* 14 *
* 15 *
* 16 *
*
* THIS CONSISTS OF FINDING WHICH UNLABELED NODE VECTOR HAS THE
* LEAST CUMULATIVE WORTH. BY MAKING WORTH A FUNCTION SUCH THAT
* THE BETTER A NODE'S WORTH, THE SMALLER THE WORTH VALUE, CUM-
* ULATIVE WORTH MUST BE ORIGINALLY SET TO INFINITY, OR IN THIS
* CASE, 32767.
*
* SET LEAST TO 32767
* DO FOR EACH NODE
*   IF NODE IS LABELED
*     cycle
*   elseif WORTH MEASURE < LEAST
*     SET LEAST TO WORTH MEASURE
*     SAVE NODE NUMBER
*   endif NODE IS LABELED
* enddo FOR EACH NODE
*
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521

```



## EXPLANATION OF PATH DETERMINATION ALGORITHM

```
#####  
# 1 OUR PROBLEM IS TO DETERMINE A ROUTE FOR A GROUND FORCE UNIT TO 1523  
# 2 FOLLOW, SUCH THAT THE PATH GETS THE UNIT TO ITS DESTINATION NODE FROM 1524  
# 3 ITS STARTING NODE AS QUICKLY AND/OR AS LEAST RISKY AS POSSIBLE. THIS 1525  
# 4 IS A SHORTEST PATH PROBLEM, WITH SOME TWISTS THRUOUT IN. ONE OF THESE 1526  
# 5 TWISTS IS THAT EACH UNIT HAS A DIFFERENT RATE OF TRAVEL, WITH THAT RATE 1527  
# 6 VARYING ACCORDING TO THE TIME OF DAY. ANOTHER IS THAT WE REALLY HAVE 1528  
# 7 MORE THAN ONE UNIT, AND WHEN MORE THAN ONE UNIT HAS THE SAME DESTINATION 1529  
# 8 NODE, WE MAY WISH FOR THE UNITS TO ARRIVE THERE AT THE SAME TIME. WE HAVE 1530  
# 9 FOUND THAT THE SIMULTANEOUS DESTINATION TIME PROBLEM MAY BE SOLVED BY STARTING 1531  
# 10 AT THE DESTINATION NODE AND WORKING OUR WAY BACK TO THE STARTING NODE, RUN- 1532  
# 11 NING TIME IN REVERSE. 1533  
# 12 PERHAPS IT WOULD BE BEST TO FIRST EXPLAIN DIJKSTRA'S ALGORITHM FOR ONE UNIT 1534  
# 13 SUPPOSE THAT THERE ARE N NODES, NUMBERED FROM 1 TO N. AND SUPPOSE FURTHER 1535  
# 14 THAT WE WISH TO FIND THE SHORTEST PATH BETWEEN NODES 1 AND N. THAT IS, NODE 1 1536  
# 15 IS THE DESTINATION, AND NODE N IS THE STARTING NODE. LABEL 1 WITH THE PERM- 1537  
# 16 ANENT VALUE ZERO, AND TENTATIVELY LABEL ALL OTHERS WITH VALUE INFINITY. EACH 1538  
# 17 TIME A PERMANENT LABEL IS ASSIGNED, IT REPRESENTS THE SHORTEST DISTANCE BETWEEN 1539  
# 18 THAT NODE AND THE DESTINATION NODE, NODE N. 1540  
# 19 ONE BY ONE, COMPARE EACH NODE LABEL EXCEPT THAT AT 1 WITH THE SUM OF THE 1541  
# 20 LABEL OF NODE 1 (THAT IS, 0) AND THE DIRECT DISTANCE FROM NODE 1 TO THE NODE 1542  
# 21 IN QUESTION. THE SMALLER OF THE TWO NUMBERS IS THE NEW TENTATIVE LABEL. 1543  
# 22 NEXT, DETERMINE THE SMALLEST OF THE N-1 TENTATIVE LABELS AND DECLARE IT 1544  
# 23 PERMANENT. SUPPOSE THAT NODE K IS THE ONE PERMANENTLY LABELED. THEN, 1545  
# 24 ONE AT A TIME, COMPARE EACH OF THE N-2 REMAINING TENTATIVE NODE LABELS TO 1546  
# 25 THE SUM OF THE LABEL JUST ASSIGNED PERMANENTLY TO NODE K AND THE DIRECT 1547  
# 26 DISTANCE FROM NODE K TO THE NODE UNDER CONSIDERATION. THE SMALLER OF THE TWO 1548  
# 27 NUMBERS BECOMES THE TENTATIVE LABEL. DETERMINE THE MINIMUM OF THE N-2 TEN- 1549  
# 28 TATIVE LABELS, DECLARE IT PERMANENT, AND MAKE IT THE BASIS OF ANOTHER MOD- 1550  
# 29 IFICATION OF THE REMAINING TENTATIVE LABELS OF THE TYPE DESCRIBED ABOVE. 1551  
# 30 WHEN, AFTER AT MOST N-1 EXECUTIONS OF THE FUNDAMENTAL ITERATIVE STEP, NODE 1552  
# 31 N IS PERMANENTLY LABELED, THE PROCEDURE TERMINATES. 1553  
# 32 THE PATH FROM N TO 1 MAY BE RECONSTRUCTED IF, FOR EACH LABELED NODE, THE 1554  
# 33 NODE FROM WHICH IT WAS LABELED (ITS PREDECESSOR) IS RECORDED. THEN, BY START- 1555  
# 34 ING AT N, WE MAY SIMPLY FOLLOW THE PREDECESSORS BACK TO THE DESTINATION NODE. 1556  
# 35 THIS PROCESS IS CORRECT, BUT FOR EASE OF HANDLING ON THE COMPUTER, CERTAIN 1557  
# 36 ADJUSTMENTS HAVE BEEN MADE. THESE WERE ORIGINALLY SUGGESTED BY MINTY AND OTHERS 1558
```

TRW, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
LAPAS FLOW

PAGE 58.001

```
# 37      INSTEAD OF CONSIDERING A LABEL TO POSSESS A VALUE, WE SAY THAT A NODE IS EITHER  
# 38      LABELED OR IT IS NOT, AND WE CONSIDER THE DISTANCE FROM NODE 1 TO BE A SEPARATE  
# 39      ENTITY.  AND INSTEAD OF USING INFINITY, WE USE THE LARGEST NUMBER ALLOWABLE.  
# 40      WE ALSO DO SOMETHING WHICH WASN'T ANTICIPATED BY THESE GENTLEMEN, AND THAT  
# 41      IS TO SOMETIMES RUN THE ALGORITHM IN REVERSE.  NOW THE BASIC STEPS ARE STILL  
# 42      THE SAME EXCEPT THAT WE ARE RUNNING TIME IN REVERSE, USING AN ARRIVAL TIME  
# 43      AS THE STARTING POINT.  THUS, ALL DISTANCES ARE NEGATIVE, AND INSTEAD OF  
# 44      LOOKING FOR THE LEAST 'LABEL', WE LOOK FOR THE GREATEST.  BY DOING THIS,  
# 45      WE FIND OUT AT WHAT TIME A UNIT HAD TO LEAVE ITS STARTING NODE IN ORDER TO  
# 46      ARRIVE AT ITS DESTINATION NODE AT THE GIVEN TIME.  
# 47      THUS WE HAVE TWO CHOICES IN APPROACH TO THIS PROBLEM.  WE MAY WISH TO FIND  
# 48      THE SHORTEST PATH FROM NODE 1 TO NODE N, OR WE MAY FIND WHICH NODE IS SO FAR  
# 49      FROM NODE N.  
# 50      RATHER THAN WORK WITH STRICTLY DISTANCE, WE WILL REALLY BE WORKING WITH  
# 51      TIME, AND THE TWO QUESTIONS WE WILL ADDRESS WILL BE 1) HOW FAST CAN WE GET FROM  
# 52      NODE 1 TO NODE N? AND 2) GIVEN A TIME WE WISH TO ARRIVE AT NODE N, AT WHAT TIME  
# 53      SHOULD WE LEAVE NODE 1?  
#
```

#####

```
# 1559  
# 1560  
# 1561  
# 1562  
# 1563  
# 1564  
# 1565  
# 1566  
# 1567  
# 1568  
# 1569  
# 1570  
# 1571  
# 1572  
# 1573  
# 1574  
# 1575
```



## COMPUTE SOLUTION+VECTORS AND ROUTE+VECTOR

REF  
PAGE

```
*****
* 1 . SOLUTION+VECTORS ARE CONSTRUCTED BY TRACING THE PATH OF NODE+VECTORS.
* 2 . THIS IS DONE BY STARTING AT NODE I AND FOLLOWING ITS PREDECESSOR POINT-
* 3 . ER, THEN THE PREDECESSOR'S PREDECESSOR POINTER, ETC., UNTIL NODE K IS
* 4 . REACHED. EACH TIME A NEW PREDECESSOR IS REACHED, A NEW SOLUTION+VECTOR
* 5 . IS CREATED. ITS APPROPRIATE VALUES ARE ENTERED, AND IT IS LINKED TO
* 6 . ANY OTHER SOLUTION+VECTORS THAT REFER TO THE SAME NODE. AT THE SAME
* 7 . TIME AS THE SOLUTION+VECTORS ARE BEING CONSTRUCTED, THE PATH'S ROUTE+
* 8 . VECTOR IS ALSO BEING MADE. THIS IS BECAUSE MANY OF ITS ENTRIES ARE
* 9 . ACCUMULATIONS OF INFORMATION GAINED AT EACH NODE.
* 10 . ONE WORD OF NOTE, FOR UNIFORMITY, THE SOLUTION+VECTORS OF ALL FOR-
* 11 . WARD PATHS REMAIN AS THEY ARE CONSTRUCTED, BUT THE BACKWARD PATH SOL-
* 12 . UTION+VECTORS HAVE THEIR ORDER REVERSED, THUS, THERE IS NO DATA BASE
* 13 . DISTINCTION BETWEEN BACKWARD AND FORWARD PATHS.
* 14 .
* 15 . INCREMENT ROUTE NUMBER COUNTER
* 16 . SET THE UNIT NUMBER
* 17 . SET TOTAL ROUTE TIME TO 0
* 18 . SET TOTAL RISK MEASURE TO 0
* 19 . SET NUMBER OF NODES TO 1
* 20 . SET ROUTE MOVEMENT TYPE
* 21 . SET THE STARTING TIME
* 22 . SET J TO I
* 23 . DO UNTIL NODE K IS REACHED
* 24 .   INCREMENT SOLUTION+VECTOR COUNTER
* 25 .   SET NODE NUMBER OF SOLUTION+VECTOR TO J
* 26 .   SET LANE NUMBER AND PARK TIME
* 27 .   SET CODE FOR CONFLICT RESOLUTION CONSIDERATION
* 28 .   INCREMENT ROUTE DISTANCE BY LINK DISTANCE FROM J TO PREDECESSOR NODE
* 29 .   INCREMENT TOTAL ROUTE TIME BY LINK TIME VALUE
* 30 .   INCREMENT TOTAL RISK BY LINK RISK VALUE
* 31 .   SET TIME IN ACCORDING TO ROUTE TYPE (FORWARD OR BACKWARD)
* 32 .   SET TIME OUT ACCORDING TO ROUTE TYPE
*****
```

1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608





CALCULATE SECOND BEST PATH

REF  
PAGE

```

*****
* 1 . THIS IS DONE BY SYSTEMATICALLY "ERASING" NODES ALONG THE BEST PATH
* 2 .AND CALCULATING THE NEW BEST PATH. AFTER THE NEW PATH IS CALCULATED,
* 3 .THE ERASED NODE IS RESTORED, AND THE NEXT ERASED, AND SO ON. THE NEW
* 4 .PATH WITH THE COMPLETION TIME CLOSEST TO THE INITIAL BEST PATH IS THEN
* 5 .THE SECOND BEST PATH. FOR ADDED INSIGHT, WE WILL KEEP THE NEXT THREE
* 6 .BEST PATHS, HOWEVER, IT IS OFTEN THE CASE THAT TWO (OR EVEN ALL THREE)
* 7 .OF THE NEXT BEST PATHS ARE IDENTICAL.
* 8
* 9 .do UPON FUNCTION REQUEST
* 10 .PURGE ALL ROUTES
* 11
* 12 .THIS ROUTINE CREATES THREE FICTITIOUS UNITS TO BE ASSOCIATED WITH THE
* 13 .THREE NEW CALCULATED ROUTES. CHECK FOR STORAGE ROOM IN THE UNITS DATA
* 14 .BASE.
* 15
* 16 .if THERE IS NOT ENOUGH ROOM IN UNITS FOR THREE MORE UNIT+VECTORS
* 17 .PRINT ERROR MESSAGE
* 18 .return TO CALLING PROGRAM
* 19 .endif THERE IS NOT ENOUGH ROOM IN UNITS FOR THREE MORE UNIT+VECTORS
* 20 .PROMPT FOR UNIT NAME
* 21 .if INPUT IS NULL
* 22 .return TO CALLING PROGRAM
* 23 .endif INPUT IS NULL
* 24 .if ENTERED UNIT NAME IS NOT IN THE DATA BASE
* 25 .PRINT ERROR MESSAGE
* 26 .TRY AGAIN
* 27 .else ENTERED UNIT NAME IS LISTED
* 28 .PROMPT FOR STARTING TIME
* 29 .PROMPT FOR RISK AND TIME PARAMETERS ...C AND K
* 30 .if DESTINATION IS FEBA AND NONE HAS BEEN DEFINED
* 31 .PRINT ERROR MESSAGE
* 32 .return TO CALLING PROGRAM
*****

```

```

* 33 *
* 34 *
59 * 35 *
* 36 *
* 37 *
* 38 *
* 39 *
* 40 *
* 41 *
55 * 42 *
* 43 *
* 44 *
* 45 *
* 46 *
* 47 *
* 48 *
* 49 *
* 50 *
* 51 *
* 52 *
* 53 *
* 54 *
* 55 *
* 56 *
55 * 57 *
* 58 *
59 * 59 *
* 60 *
* 61 *
* 62 *
* 63 *
* 64 *
* 65 *
* 66 *
* 67 *
* 68 *
*

endif DESTINATION IS FEBA
CALCULATE BEST PATH ...FORWARD MOVEMENT WITH NO DECONFLICTING
COMPUTE SOLUTION+VECTORS AND ROUTE+VECTOR
. NEW SYSTEMATICALLY ERASE NODES AND CALCULATE NEW PATHS
. INITIALIZE ARRAY TO CONTAIN THREE NEAREST TIMES TO 32767
do FOR SECOND THROUGH NEXT-TO-LAST NODES OF THE BEST PATH
  ERASE ROADS LEADING TO THE NODE
  REINITIALIZE 'WORKING LIST' NODE+VECTOR ENTRIES
  CALCULATE BEST PATH ...FORWARD WITH NO DECONFLICTING
  COMPARE COMPLETION TIME WITH OTHERS IN TIME ARRAY
  IF THIS TIME IS LESS THAN ANY ONE TIME IN ARRAY
    PLACE THIS TIME IN ARRAY
    ORDER ARRAY IN ASCENDING FASHION
    REMEMBER ERASED NODE
  endif THIS TIME IS LESS THAN ANY ONE TIME IN ARRAY
  RESTORE ERASED NODE'S ROADS
enddo FOR SECOND THROUGH NEXT-TO-LAST NODES OF THE BEST PATH
. CALCULATE THE THREE NEXT BEST PATHS AND ROUTES
. do FOR EACH ENTRY IN TIME ARRAY
  ERASE REMEMBERED NODE
  REINITIALIZE 'WORKING LIST' NODE+VECTOR ENTRIES
  CALCULATE BEST PATH ...FORWARD WITH NO DECONFLICTING
  COMPUTE SOLUTION+VECTORS AND ROUTE+VECTOR
  CREATE NEW UNIT+VECTOR
  ASSOCIATE NEW UNIT+VECTOR WITH THIS ROUTE
  PRINT MAP+NUMBER, NODE+NUMBER OF ERASED NODE
  PRINT UTM LATITUDE, LONGITUDE OF ERASED NODE
  PRINT DELAY TIME
  RESTORE ERASED NODE
enddo FOR EACH ENTRY IN TIME ARRAY
endif ENTERED UNIT NAME IS NOT IN THE DATA BASE
enddo UPON FUNCTION REQUEST

```

1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713



FIND THE ORDER OF PRIORITIES FOR BEST NETWORK TRAVEL

REF  
PAGE

```

*****
* 1 THIS PROBLEM IS: GIVEN N UNITS TRAVELING THROUGH A ROAD NETWORK, AND
* 2 GIVEN THAT WHEN ANY TWO UNITS WISH TO USE THE SAME NODE DURING OVER-
* 3 LAPPING TIME INTERVALS THE ONE WITH THE HIGHER PRIORITY USES IT FIRST
* 4 WHILE THE OTHER WAITS OR GOES AROUND, WHAT ORDERING OF PRIORITIES PRO-
* 5 DUCES THE BEST RESULTS?
* 6 THIS ROUTINE ANSWERS THIS QUESTION BY TRYING EVERY POSSIBLE COMBINA-
* 7 TION OF PRIORITY ORDERINGS, AND SEEING WHICH ONE DOES IN FACT PRODUCE
* 8 THE BEST RESULTS. BY 'BEST' WE MEAN THAT THE EARLIEST START TIME OF
* 9 THE UNITS IN THE CHOSEN PRIORITY SCHEME IS LATER THAN ALL OTHER EAR-
* 10 LIEST TIMES PRODUCED USING THE OTHER PRIORITY SCHEMES.
* 11
* 12 do UPON FUNCTION REQUEST
* 13 PURGE ALL ROUTES
* 14 PROMPT FOR TIME MOVEMENT ...FORWARD GIVES RATHER UNINTERESTING RESULTS
* 15 if MOVEMENT IS FORWARD
* 16 PROMPT FOR STARTING TIME
* 17 else MOVEMENT IS BACKWARD
* 18 if DESTINATION IS THE FEBA AND A FEBA HAS NOT BEEN DEFINED
* 19 PRINT ERROR MESSAGE
* 20 TRY AGAIN
* 21 endif DESTINATION IS THE FEBA
* 22 PROMPT FOR ARRIVAL TIME
* 23 endif MOVEMENT IS FORWARD
* 24
* 25 .OBTAIN UNIT NAMES. AT THIS TIME, THIS ROUTINE WILL HANDLE AT MOST FIVE
* 26 .UNITS. THIS IS BECAUSE OF TIME CONSIDERATIONS.
* 27
* 28 do AT MOST FIVE TIMES
* 29 PROMPT FOR UNIT NAME
* 30 if INPUT IS NULL AND NO UNIT NAME HAS BEEN ENTERED
* 31 return TO CALLING PROGRAM
* 32 endif INPUT IS NULL AND NO UNIT NAME HAS BEEN ENTERED
*****
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746

```

```
* 33
* 34
* 35
* 36
* 37
* 38
* 39
* 40
* 41
* 42
* 43
* 44
* 45
* 46
* 47
* 48
* 49
* 50
* 51
* 52
* 53
* 54
* 55
* 56
* 57
* 58
* 59
* 60
* 61
* 62
* 63
* 64
* 65
* 66
* 67
* 68
* 69
* 70

if INPUT IS NULL AND AT LEAST ONE UNIT NAME HAS BEEN ENTERED
  undo AT MOST FIVE TIMES
else INPUT IS NOT NULL
  if ENTERED UNIT NAME IS NOT IN THE DATA BASE
    PRINT MESSAGE
    TRY AGAIN
  endif ENTERED UNIT NAME IS NOT IN THE DATA BASE
  REMEMBER UNIT NAME
  endif INPUT IS NULL AND AT LEAST ONE UNIT NAME HAS BEEN ENTERED
enddo AT MOST FIVE TIMES

. GENERATE ALL POSSIBLE PRIORITY COMBINATIONS AND MAKE PATH CALCULATIONS
. USING TIME.

. INITIALIZE EARLIEST TIME
do FOR EACH COMBINATION OF PRIORITIES
  RESET NONE+VECTOR SOLUTION+VECTOR POINTERS
  RESET THIS COMBINATION'S EARLIEST START TIME
  do FOR EACH UNIT
    CALCULATE BEST PATH ....WITH DECONFLICTING
    MAKE SOLUTION+VECTORS AND ROUTE+VECTOR
    if THE START TIME IS EARLIER THAN THE OTHERS
      REMEMBER THE TIME
    endif THIS PATH WORSE THAN OTHERS
  enddo FOR EACH UNIT
  if EARLIEST START TIME IS LATER THAN ALL OTHER EARLIEST TIMES
    REMEMBER PRIORITY COMBINATION
    REMEMBER EARLIEST START TIME
  endif EARLIEST START TIME IS LATER THAN ALL OTHER EARLIEST TIMES
enddo FOR EACH COMBINATION OF PRIORITIES

. CALCULATE RESULTS

. REINITIALIZE SOLUTION+VECTORS AND ROUTE+VECTORS
do FOR EACH UNIT USING THE BEST PRIORITY SCHEME
  CALCULATE BEST PATH
  COMPUTE SOLUTION+VECTORS AND ROUTE+VECTOR
enddo FOR EACH UNIT USING THE BEST PRIORITY SCHEME
```

```
* 1747
* 1748
* 1749
* 1750
* 1751
* 1752
* 1753
* 1754
* 1755
* 1756
* 1757
* 1758
* 1759
* 1760
* 1761
* 1762
* 1763
* 1764
* 1765
* 1766
* 1767
* 1768
* 1769
* 1770
* 1771
* 1772
* 1773
* 1774
* 1775
* 1776
* 1777
* 1778
* 1779
* 1780
* 1781
* 1782
* 1783
* 1784
```



TRW, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
LAMAS FLOW

PAGE 61.002

\* 71  
\* 72  
\*

PRINT MESSAGE INFORMING USER THAT RESULTS ARE AVAILABLE  
enddo UPON FUNCTION REQUEST

\* 1785  
\* 1786  
\*

\*\*\*\*\*

CALCULATE BEST NODE AT WHICH TO INTERDICT

REF  
PAGE

```

1  . THIS ROUTINE CALCULATES THE NODE AT WHICH IT IS BEST TO INTERDICT,
2  .WHERE 'BEST' MEANS 'HAS THE GREATEST NORTH DIFFERENCE BETWEEN THE BEST
3  .PATH AND THE PATH WITH INTERDICTION'. IN ADDITION, THE PROGRAM FINDS
4  .THE TIME WINDOW DURING WHICH THE INTERDICTION SHOULD OCCUR.
5  . THIS IS DONE IN A MANNER SIMILAR TO THE PROCESS FOR CALCULATING THE
6  .SECOND BEST PATH. EACH NODE OF THE BEST PATH IS SYSTEMATICALLY ER-
7  .ASED, BUT THEN THERE IS A DIFFERENCE. THIS ROUTINE FORCES THE UNIT TO
8  .TRAVEL A DISTANCE DOWN THE BEST PATH, AND THEN LETS IT GO ITS OWN WAY.
9  .E.G., SUPPOSE THAT THE FIFTH NODE HAS BEEN ERASED, THERE WILL BE FOUR
10 .PATHS CALCULATED FOR THIS ONE NODE ERASURE. FIRST, THE UNIT WILL BE
11 .FORCED TO TRAVEL THROUGH NODE ONE, AND THEN IT MAY GO AS IT PLEASES.
12 .NEXT, IT IS FORCED THROUGH NODE 2, THEN THROUGH NODE 3, AND FINALLY,
13 .THROUGH NODE FOUR. IN THIS WAY, WE SIMULATE TIME RELATED INTERDICTION.
14
15 do UPON FUNCTION REQUEST
16   PURGE ALL ROUTES
17
18   . WE WILL ACTUALLY KEEP THE THREE BEST INTERDICTION CASES IN AN ARRAY.
19   .AND, WE WILL CREATE THREE FICTITIOUS UNITS.
20
21   INITIALIZE BEST ARRAY
22   if THERE IS NOT ROOM FOR THREE MORE UNIT+VECTORS
23     PRINT MESSAGE
24     return TO CALLING PROGRAM
25   endif THERE IS NOT ROOM FOR THREE MORE UNIT+VECTORS
26   PROMPT FOR UNIT NAME
27   if INPUT IS NULL
28     return TO CALLING PROGRAM
29   else INPUT IS NOT NULL
30     if UNIT NAME IS NOT IN DATA BASE
31       PRINT MESSAGE
32       TRY AGAIN

```

1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819



```

* 33      endif UNIT NAME IS NOT IN DATA BASE
* 34      endif INPUT IS NULL
* 35      .GET PARAMETERS
* 36      .
* 37      PROMPT FOR START TIME
* 38      PROMPT FOR RISK AND TIME PARAMETERS ...C AND K
* 39      CALCULATE BEST PATH
* 40      COMPUTE SOLUTION+VECTORS AND ROUTE+VECTOR
59 * 41
* 42      .SYSTEMATICALLY ERASE NODES, AND FORCE UNIT TO TRAVEL
* 43      .
* 44      do FOR THE SECOND THROUGH NEXT-TO-LAST NODES OF THE BEST PATH
* 45      ERASE NODE
* 46      do FOR EACH NODE FROM FIRST NODE TO PREDECESSOR OF ERASED NODE
* 47      LOOK IN SOLUTION+VECTOR OF NODE TO OBTAIN START TIME
* 48      .
* 49      .WE'LL ONLY CALCULATE FROM THIS NODE TO THE DESTINATION
* 50      .
* 51      CALCULATE BEST PATH OF THIS NODE TO THE DESTINATION
* 52      IF PATH'S WORTH IS GREATER THAN ANY OF THOSE IN THE ARRAY
* 53      PLACE THIS WORTH IN NORTH ARRAY
* 54      REMEMBER INTERDICTIONED NODE
* 55      REMEMBER 'FORCED-TO' NODE
* 56      endif PATH'S WORTH IS GREATER THAN ANY OF THOSE IN THE ARRAY
* 57      enddo FOR EACH NODE FROM FIRST NODE TO PREDECESSOR OF ERASED NODE
* 58      RESTORE ERASED NODE
* 59      enddo FOR THE SECOND THROUGH NEXT-TO-LAST NODE OF BEST PATH
* 60      .
* 61      .CALCULATE CHOSEN PATHS. EACH PATH WILL BE THE COMBINATION OF TWO
* 62      .PATHS, ONE FROM THE FIRST NODE TO THE INTERDICTIONED NODE, THEN ANOTHER
* 63      .FROM THE INTERDICTIONED NODE TO THE DESTINATION.
* 64      .
* 65      do FOR EACH ENTRY IN NORTH ARRAY
* 66      ERASE REMEMBERED NODE
* 67      INITIALIZE 'WORKING LIST' NODE+VECTOR ENTRIES
* 68      CALCULATE BEST PATH FROM START NODE TO PREDECESSOR OF ERASED NODE
* 69      COMPUTE SOLUTION+VECTORS AND ROUTE+VECTOR
59 * 70

```

```

1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857

```

TRU, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
LOHAS FLOW

PAGE 62.002

```
* /1 INITIALIZE 'MARKING LIST' NODE<VECTOR ENTRIES 1858
* /2 CALCULATE BEST PATH FROM PREDECESSOR TO DESTINATION NODE 1859
59 * /3 COMPUTE SOLUTION<VECTORS AND ROUTE<VECTOR 1860
* /4 COMBINE SOLUTION<VECTORS AND ROUTE<VECTORS TO MAKE ONE ROUTE 1861
* /5 CREATE NEW UNIT ASSOCIATED WITH THIS ROUTE 1862
* /6 PRINT MAP<NUMBER, NODE<NUMBER OF INTERDICTED NODE 1863
* /7 PRINT MAP<NUMBER, NODE<NUMBER OF 'FORCED' NODE 1864
* /8 PRINT TIME WINDOW 1865
* /9 PRINT TIME DELAY 1866
* 80 enddo FOR EACH ENTRY IN WORTH ARRAY 1867
* 81 return TO CALLING PROGRAM 1868
* 82 enddo UPON FUNCTION REQUEST 1869
*
```

100 MANY LINES IN SEGMENT



TRK, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
LANAS FLUX

PAGE 63

PRESENT RESULTS

REF  
PAGE

```
*****
* 1 . THESE ROUTINES ENABLE THE USER TO SEE THE RESULTS OF THE PATH ALGO-
* 2 . RITHMS AS PRINTED OUTPUT.
* 3 .
* 4 do UPON FUNCTION REQUEST
* 5   do FOREVER
* 6     DISPLAY FUNCTION MENU
* 7     WAIT FOR USER INPUT
* 8     do CASE OF
* 9       TABLE:
*10
*11 HARDCOPY:
*12
*13 EXIT:
*14   return TO CALLING PROGRAM
*15   enddo CASE OF
*16   enddo FOREVER
*17   enddo UPON FUNCTION REQUEST
*****
```

PRINT TABLE OF ROUTE NUMBERS WITH ASSOCIATED UNIT NAMES

PRINT PATH STATISTICS

return TO CALLING PROGRAM

enddo CASE OF

enddo FOREVER

enddo UPON FUNCTION REQUEST

1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887

PRINT TABLE OF ROUTE NUMBERS WITH ASSOCIATED UNIT NAMES

REF  
PAGE

```
*****
* 1 . IF THIS FUNCTION IS REQUESTED, A TABLE SHOWING ALL ROUTE NUMBERS, EACH
* 2 . WITH THE UNIT NAME WITH WHICH IT IS ASSOCIATED, WILL BE PRINTED.
* 3 .
* 4 . DO UPON FUNCTION REQUEST
* 5 . PRINT TABLE OF ROUTE NUMBERS AND ASSOCIATED UNIT NAMES
* 6 . RETURN TO CALLING PROGRAM
* 7 . ENDDO UPON FUNCTION REQUEST
*
*****
1889
1890
1891
1892
1893
1894
1895
*****
```



AD-A069 029

TRW DEFENSE AND SPACE SYSTEMS GROUP REDONDO BEACH CALIF  
LAMAS SYSTEM MANUAL. (U)  
FEB 78

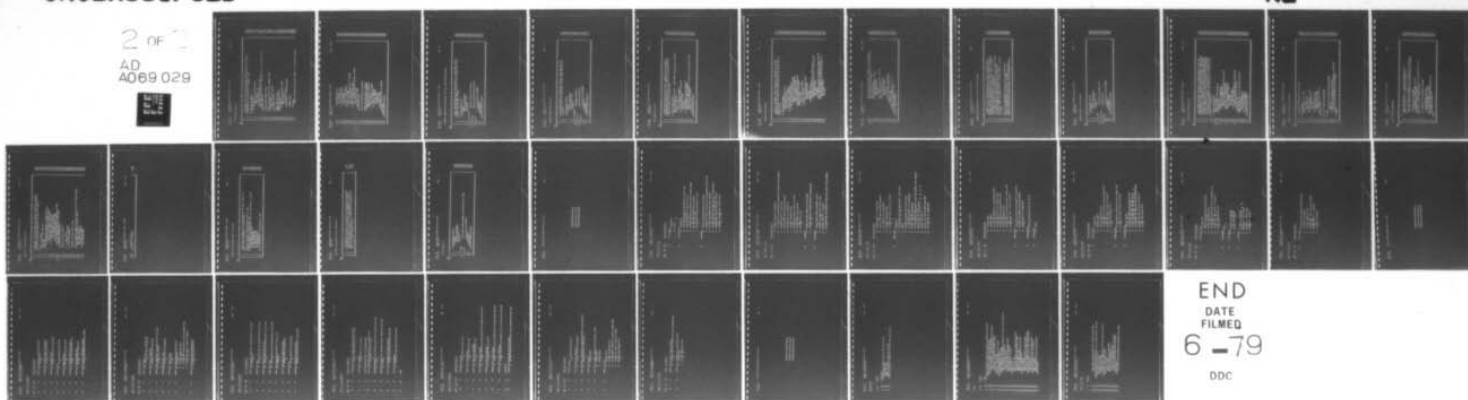
F/G 9/2

DAAG39-77-C-0112

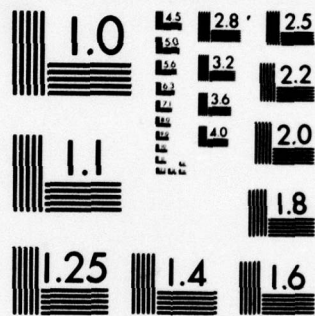
UNCLASSIFIED

NL

2 OF 2  
AD  
A069 029



END  
DATE  
FILMED  
6-79  
DDC



MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A



TRM, INC.  
21 FEB 70

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
LAWRENCE FLIN

PAGE 65

# PRINT PATH STATISTICS

REF  
PAGE

```
1 * THIS ENABLE THE USER TO HAVE HARD-COPY EVIDENCE OF THE CALCULATIONS
2 * PERFORMED BY THIS SYSTEM. THIS INCLUDES TOTAL PATH STATISTICS, AS WELL
3 * AS THE ABILITY TO HAVE OUTPUT IN A SNAPSHOT MODE.
4 *
5 * do OPEN FUNCTION REQUEST
6 *   do FOR EACH UNIT
7 *     if THERE IS NO ASSOCIATED ROUTE
8 *       PRINT MESSAGE
9 *       cycle FOR EACH UNIT
10 *     endif THERE IS NO ASSOCIATED ROUTE
11 *     PRINT TOTAL PATH STATISTICS ...DISTANCE, TIME, START AND FINISH, ETC.
12 *     enddo FOR EACH UNIT
13 *
14 * .NOW SEE ABOUT SNAPSHOT MODE
15 *
16 * PROMPT FOR USER DESIRE TO HAVE SNAPSHOT OUTPUT
17 * if INPUT IS NULL
18 *   return TO CALLING PROGRAM
19 * endif INPUT IS NULL
20 *
21 * PROMPT FOR START TIME
22 * PROMPT FOR TIME INCREMENT
23 * do FOR EACH UNIT
24 *   if THERE IS NO ASSOCIATED ROUTE
25 *     PRINT MESSAGE
26 *     cycle FOR EACH UNIT
27 *   endif THERE IS NO ASSOCIATED ROUTE
28 *
29 * .SNAPSHOT MODE
30 *
31 * do FOREVER
32 *   FIND UNIT POSITION FOR TIME OF INTEREST ...AS IN SNAPSHOT DISPLAY
```

1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928

```

* 33      if UNIT HAS NOT STARTED
* 34      PRINT MESSAGE
* 35      cycle FOR EACH UNIT
* 36      elseif UNIT HAS FINISHED
* 37      PRINT MESSAGE
* 38      cycle FOR EACH UNIT
* 39      endif UNIT HAS NOT STARTED
* 40      if UNIT IS AT A NODE
* 41      PRINT STATISTICS FOR UNIT AT THE NODE
* 42      cycle FOR EACH UNIT
* 43      else UNIT IS BETWEEN TWO NODES
* 44      PRINT STATISTICS FOR UNIT BETWEEN THE NODES
* 45      cycle FOR EACH UNIT
* 46      endif UNIT IS AT A NODE
* 47      enddo FOR EACH UNIT
* 48
* 49      .SEE IF THE USER WISHES TO CONTINUE
* 50
* 51      PROMPT FOR USER DESIRE TO CONTINUE
* 52      if USER DOES NOT WANT TO CONTINUE
* 53      PROMPT FOR NEW TIME INCREMENT
* 54      if INPUT IS NULL
* 55      return TO CALLING PROGRAM
* 56      else INPUT IS NOT NULL
* 57      SET TIME INCREMENT
* 58      INCREMENT TIME
* 59      cycle FOREVER
* 60      endif INPUT IS NULL
* 61      endif USER DOES NOT WANT TO CONTINUE
* 62      enddo FOREVER
* 63      enddo UPON FUNCTION REQUEST

```

```

1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959

```



IMPLEMENT ALGORITHMS AND FUNCTIONS USING THE CROSS-COUNTRY DATA BASE

REF  
PAGE

```

1  . THIS IS THE OTHER HALF OF THE LANAS SYSTEM. IT IS MUCH MORE LIMITED
2  . IN SCOPE THAN THE ROAD NETWORK SECTION, AND THIS IS MAINLY DUE TO THE
3  . NATURE OF THE DATA BASE INVOLVED.
4  .
5  . do UPON FUNCTION REQUEST
6  .   do FOREVER
7  .     DISPLAY FUNCTION MENU
8  .     WAIT FOR USER INPUT
9  .     do CASE OF
10 .       INITIALIZE:
11 .         PERFORM TERRAIN INITIALIZATION
12 .       PATHS:
13 .         PERFORM TERRAIN PATH CALCULATIONS
14 .       EXIT:
15 .         return TO CALLING PROGRAM
16 .         enddo CASE OF
17 .         enddo FOREVER
18 .       enddo UPON FUNCTION REQUEST

```

1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978

PERFORM TERRAIN INITIALIZATION

REF  
PAGE

```

1  . IN THE CROSS-COUNTRY AND CONCEALMENT SECTION, THERE ARE ONLY TWO
2  . THINGS WHICH NEED TO BE INITIALIZED, THE MAP+NUMBER, AND THE UNITS
3  . DATA BASE.
4
5  do UPON FUNCTION REQUEST
6  do FOREVER
7  DISPLAY FUNCTION MENU
8  WAIT FOR USER INPUT
9  do CASE OF
10 MAP+NUMBER:
11   MAP+NUMBER TO BE USED
12 UNITS:
13   ESTABLISH UNIT+VECTORS
14 EXIT:
15   return TO CALLING PROGRAM
16   enddo CASE OF
17   enddo FOREVER
18   enddo UPON FUNCTION REQUEST

```

1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997



TRU, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
LSMAS FLOW

PAGE 68

MAP<NUMBER TO BE USED

REF  
PAGE

```
1  . FOR THE CROSS-COUNTRY AND CONCEALMENT ALGORITHMS, ONLY ONE MAP MAY
2  .BE CONSIDERED AT ONE TIME. THIS IS SO BECAUSE OF THE LARGE NUMBER OF
3  .NODES WHICH ONE MAP REPRESENTS (9186), AND THE TIME INVOLVED FOR A
4  .PATH CALCULATION USING THAT MANY NODES.
5  .
6  . CO UPON FUNCTION REQUEST
7  . READ 'COMDIR.DAT' INTO MEMORY ...CROSS-COUNTRY AND CONCEALMENT DIRECTORY
8  . PROMPT FOR MAP<NUMBER
9  . IF INPUT IS NULL
10 .   RETURN TO CALLING PROGRAM
11 . ELSE IF INPUT NUMBER IS NOT IN DIRECTORY
12 .   PRINT ERROR MESSAGE
13 .   TRY AGAIN
14 . ELSE INPUT NUMBER IS IN THE DIRECTORY
15 .   READ THE MAP'S NODES INTO MEMORY
16 .   ESTABLISH THE SCREEN EXTREMES ...VALUES ARE IN THE DIRECTORY ENTRY
17 .   RETURN TO CALLING PROGRAM
18 . ENDIF INPUT IS NULL
19 . ENDDO UPON FUNCTION REQUEST
```

1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017

TRW, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
LANAS FLOW

PAGE 69

ESTABLISH UNIT-VECTORS

REF  
PAGE

```

1  * JUST AS IN THE ROAD NETWORK SECTION, THIS DATA BASE MUST BE CON-
2  * STRUCTED BY THE USER. THE ONLY DIFFERENCE BETWEEN THE TWO IS THAT
3  * HERE, THE START AND DESTINATION ARE DESCRIBED AS UTM LATITUDE AND LONG-
4  * TITUDE PAIRS.
5  *
6  * do UPON FUNCTION REQUEST
7  *   if NO MAP HAS BEEN INITIALIZED
8  *     PRINT ERROR MESSAGE
9  *     return TO CALLING PROGRAM
10 *   else A MAP HAS BEEN INITIALIZED
11 *     do UNTIL UNIT-COUNTER EQUALS 60
12 *       if USER DESIRES RE-INITIALIZATION
13 *         CLEAR ALL EXISTING UNIT-VECTORS
14 *         RESET UNIT-COUNTER TO ZERO
15 *       endif USER DESIRES RE-INITIALIZATION
16 *       PROMPT FOR ENTRY OF UNIT NAME
17 *       if INPUT IS NULL
18 *         return TO CALLING PROGRAM
19 *       else INPUT IS NOT NULL
20 *         if THIS UNIT NAME ALREADY EXISTS
21 *           PRINT ERROR MESSAGE
22 *           TRY AGAIN
23 *         else UNIT NAME IS NEW
24 *           PROMPT FOR UTM COORDINATES OF STARTING NODE
25 *           if COORDINATES DO NOT LIE WITHIN THE INITIALIZED MAP
26 *             PRINT ERROR MESSAGE
27 *             TRY AGAIN
28 *           endif COORDINATES DO NOT LIE WITHIN THE INITIALIZED MAP
29 *           PROMPT FOR UTM COORDINATE OF DESTINATION NODE
30 *           if COORDINATES DO NOT LIE WITHIN THE INITIALIZED MAP
31 *             PRINT ERROR MESSAGE
32 *             TRY AGAIN

```



TRG, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
LAMAS FLOW

PAGE 69.001

```
* 33
* 34
* 35
* 36
* 37
* 38
* 39
* 40
* 41
* 42
* 43
* 44
* 45
* 46
* 47
* 48
* 49
* 50
* 51
* 52
* 53
* 54
* 55
* 56
* 57
* 58
* 59
* 60
* 61
* 62
* 63
* 64
* 65
* 66
* 67
* 68
* 69

endif COORDINATES DO NOT LIE WITHIN THE INITIALIZED MAP
CHANGE ENTERED COORDINATES TO AN INDEX INTO THE DATA BASE
MAKE ASSIGNMENTS TO UNIT VECTOR
PROMPT FOR PRIORITY
if PRIORITY NOT BETWEEN 0 AND 127
  PRINT ERROR MESSAGE
  TRY AGAIN
endif PRIORITY NOT BETWEEN 0 AND 127
PROMPT FOR TYPE CODE
if TYPE CODE NOT ALLOWABLE
  PRINT ERROR MESSAGE
  TRY AGAIN
endif TYPE CODE NOT ALLOWABLE
endif THIS UNIT NAME ALREADY EXISTS
endif INPUT IS NULL
enddo UNTIL UNIT COUNTER EQUALS SIXTY
endif NO MAP HAS BEEN INITIALIZED
enddo UPON FUNCTION REQUEST
```

TRU, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
LAKES FLIN

PAGE 70

CHANGE ENTERED COORDINATES TO AN INDEX INTO THE DATA BASE

REF  
PAGE

- 1 .IN ORDER TO KEEP THE MEMORY USAGE AT A MINIMUM, AN INDEXING SCHEME
- 2 .HAS BEEN DEvised. THE DATA BASE HAS BEEN PACKED TOGETHER SO THAT
- 3 .ONE WORD CONTAINS FOUR DATA VALUES, OR EVERY DATA VALUE IS REPRESENTED
- 4 .BY FOUR BITS. ONE FOUR-BIT AREA IS CALLED A NIBBLE (HALF OF A BYTE).
- 5 .THUS, THERE ARE 8184 NIBBLES IN THE DATA BASE.
- 6 . THE POINT IS THEN TO MAP THE UTM COORDINATE PAIR TO ONE SQUARE ON THE
- 7 .93 X 89 GRID, AND THEN TRANSLATE THAT VALUE TO A NIBBLE VALUE.
- 8 .THE POSSIBLE RANGE OF THE UTM COORDINATE PAIRS IS 1 TO 505
- 9 .(22 \* 23, = SQUARE KILOMETERS ON A MAP), SO WE'RE MAPPING 1 TO 506 TO
- 10 .1 TO 8184.
- 11 .
- 12 .CALCULATE THE VALUE BETWEEN 1 AND 506 FOR THE GIVEN COORDINATES
- 13 .CALCULATE THE PERCENTAGE OF THIS VALUE OF THE WHOLE
- 14 .MULTIPLY THE PERCENTAGE BY 8183 AND ADD 1 TO OBTAIN THE NIBBLE
- 15 .ASSIGN THE VALUE TO EITHER STARTING LOCATION OR DESTINATION LOCATION

2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084



## PERFORM TERRAIN PATH CALCULATIONS

REF	PAGE
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	22
23	23
24	24
25	25
26	26
27	27
28	28
29	29
30	30
31	31
32	32
33	33
34	34
35	35
36	36
37	37
38	38
39	39
40	40
41	41
42	42
43	43
44	44
45	45
46	46
47	47
48	48
49	49
50	50
51	51
52	52
53	53
54	54
55	55
56	56
57	57
58	58
59	59
60	60
61	61
62	62
63	63
64	64
65	65
66	66
67	67
68	68
69	69
70	70
71	71
72	72
73	73
74	74
75	75
76	76
77	77
78	78
79	79
80	80
81	81
82	82
83	83
84	84
85	85
86	86
87	87
88	88
89	89
90	90
91	91
92	92
93	93
94	94
95	95
96	96
97	97
98	98
99	99
100	100

1	do	UPON FUNCTION REQUEST	2086
2	do	FOREVER	2087
3		DISPLAY FUNCTION MENU	2088
4		WAIT FOR USER INPUT	2089
5	do	CASE OF	2090
6	PATH:		2091
7		CALCULATE CROSS-COUNTRY PATH	2092
8	SOLUTION:		2093
9		DISPLAY RESULTS	2094
10	EXIT:		2095
11		return TO THE CALLING PROGRAM	2096
12		enddo CASE OF	2097
13		enddo FOREVER	2098
14		enddo UPON FUNCTION REQUEST	2099

TRU, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
LAWAS FLOW

PAGE 72

CALCULATE CROSS-COUNTRY PATH

REF  
PAGE

```
1  * THIS ROUTINE CALCULATES THE PATH OF A USER-ENTERED UNIT USING THE
2  * CROSS-COUNTRY DATA BASE. THE MAJOR DIFFERENCE BETWEEN THIS ROUTINE AND
3  * THE ROUTINE FOR THE ROAD NETWORK IS THAT THERE IS NOW AN ARRAY CALLED
4  * THE WORKING LIST. PREVIOUSLY, THE WORKING LIST VARIABLES WERE STORED
5  * WITHIN THE NODE+VECTOR, BUT NOW, WITH 8184 NODES TO BE CONSIDERED, IT
6  * IS TOTALLY IMPRACTICAL TO TRY TO RETAIN SPACE FOR EACH NODE'S WORKING
7  * LIST ENTIRE. THUS, A WORKING LIST ARRAY EXISTS. IT IS TREATED AS A
8  * LIST OF VECTORS, BEING USED IN ORDER (FROM INDEX 1 TO N) AS THEY ARE
9  * NEEDED BY THE ALGORITHM.
10
11  * CO UPON FUNCTION REQUEST
12  * PURGE ALL ROUTES
13  * OBTAIN UNIT NAMES TO BE CONSIDERED
14  * DO FOR EACH UNIT
15  *   OBTAIN THE PATH PARAMETERS
16  *   IF THERE IS AN ERROR RETURN
17  *   GO BACK TO OBTAIN UNIT NAMES AND TRY AGAIN
18  *   ENDIF THERE IS AN ERROR RETURN
19  *   CALCULATE THE CROSS-COUNTRY PATH
20  *   IF THE PATH COULD NOT COMPLETE
21  *     PRINT MESSAGE
22  *     cycle FOR EACH UNIT
23  *   ELSE THE PATH DID COMPLETE
24  *     CONSTRUCT SOLUTION+VECTORS AND ROUTE+VECTOR
25  *   ENDIF THE PATH COULD NOT COMPLETE
26  *   ENDDO FOR EACH UNIT
27  *   PROMPT FOR USER WISH FOR SOLUTION+VECTOR PRINTOUT
28  *   IF USER WANTS OUTPUT
29  *     PRINT SOLUTION+VECTORS
30  *   ENDIF USER WANTS OUTPUT
31  *   RETURN TO CALLING PROGRAM
32  *   ENDDO UPON FUNCTION REQUEST
33
```

2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132



TRU, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
LOGS FLOW

PAGE 73

OBTAIN UNIT NAMES TO BE CONSIDERED

REF  
PAGE

```
1 1 . THE USER MAY ENTER AT MOST SIXTY NAMES.  
2 2  
3 3  
4 4  
5 5  
6 6  
7 7  
8 8  
9 9  
10 10  
11 11  
12 12  
13 13  
14 14  
15 15  
16 16  
17 17  
18 18  
19 19  
20 20  
21 21  
22 22  
23 23  
24 24  
25 25  
26 26  
27 27  
28 28  
29 29  
30 30  
31 31  
32 32  
33 33  
34 34  
35 35  
36 36  
37 37  
38 38  
39 39  
40 40  
41 41  
42 42  
43 43  
44 44  
45 45  
46 46  
47 47  
48 48  
49 49  
50 50  
51 51  
52 52  
53 53  
54 54  
55 55  
56 56  
57 57  
58 58  
59 59  
60 60  
61 61  
62 62  
63 63  
64 64  
65 65  
66 66  
67 67  
68 68  
69 69  
70 70  
71 71  
72 72  
73 73  
74 74  
75 75  
76 76  
77 77  
78 78  
79 79  
80 80  
81 81  
82 82  
83 83  
84 84  
85 85  
86 86  
87 87  
88 88  
89 89  
90 90  
91 91  
92 92  
93 93  
94 94  
95 95  
96 96  
97 97  
98 98  
99 99  
100 100
```

TRG, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
LAPAS FLOW

PAGE 14

OBTAIN THE PATH PARAMETERS

REF  
PAGE

```
1 do
2   PROMPT FOR MOVEMENT TYPE ...BACKWARDS OR FORWARDS IN TIME
3   PROMPT FOR START TIME ...DEPENDENT UPON MOVEMENT TYPE
4
5   .REINITIALIZATION OF THE WORKING LIST IS NEEDED ONLY IF EITHER THE
6   .UNIT TYPE, START TIME, MOVEMENT TYPE, OR START LOCATION HAS CHANGED.
7
8   if REINITIALIZATION IS NEEDED
9     SET WORKING LIST ARRAY ENTRIES TO ZERO
10    and if REINITIALIZATION IS NEEDED
11      ESTABLISH THE BEGINNING NODE (<=K), AND THE DESTINATION NODE (<=I)
12      SET J TO K ...J WILL TRACE THE PATH
13
14    .SET CUMULATIVE TIME AND WORTH FOR NODE J.
15
16    if J HAS NO WORKING LIST ENTRY
17      CREATE NEW WORKING LIST ENTRY
18    end if J HAS NO WORKING LIST ENTRY
19    SET CUMULATIVE TIME TO ENTERED BEGINNING TIME
20    SET CUMULATIVE WORTH TO 0
21  enddo
```

2160  
2161  
2162  
2163  
2164  
2165  
2166  
2167  
2168  
2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2180



TRM, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
LAMAS FLOW

PAGE 75

CALCULATE THE CROSS-COUNTRY PATH

REF  
PAGE

```

1  . GIVEN NODE J, THE LAST NODE TO BE LABELED, THIS ROUTINE CALCULATES
2  . ITS ADJACENT NODES TIME VALUES. THEN THE NEXT NODE TO BE LABELED IS
3  . SELECTED AND THE PROCESS CONTINUES UNTIL NODE I IS LABELED.
4  .
5  . DO UNTIL I IS REACHED
6  .   FIND NODE J IN THE WORKING LIST
7  .   LABEL NODE J
8  .   DO FOR EACH ADJACENT NODE OF J ...EIGHT IN ALL
9  .     IF THE ADJACENT NODE HAS NO WORKING LIST ENTRY
10 .       CREATE AN ENTRY FOR THE ADJACENT NODE
11 .     ELSE THE NODE HAS A WORKING LIST ENTRY
12 .       IF THE ADJACENT NODE IS LABELED
13 .         cycle FOR EACH ADJACENT NODE
14 .       endif THE ADJACENT NODE IS LABELED
15 .     endif THE ADJACENT NODE HAS NO WORKING LIST ENTRY
16 .     CALCULATE THE ADJACENT NODE'S TIME VALUES
17 .   enddo FOR EACH ADJACENT NODE
18 .
19 . .LOOK FOR I BEING LABELED
20 .
21 . IF I IS NOW LABELED
22 .   undo UNTIL I IS REACHED
23 .   endif IS IS NOW LABELED
24 .
25 . .CALCULATE THE NEXT NODE TO LABEL
26 .
27 . FIND THE UNLABELED WORKING LIST ENTRY WHICH HAS THE LEAST NORTH MEASURE
28 . IF NONE EXISTS ...THE ROUTE COULDN'T FINISH
29 .   PRINT ERROR MESSAGE
30 .   undo UNTIL I IS REACHED
31 .   else AN ENTRY WAS FOUND
32 .     SET J TO THIS ENTRY

```

2182  
2183  
2184  
2185  
2186  
2187  
2188  
2189  
2190  
2191  
2192  
2193  
2194  
2195  
2196  
2197  
2198  
2199  
2200  
2201  
2202  
2203  
2204  
2205  
2206  
2207  
2208  
2209  
2210  
2211  
2212  
2213

TRW, INC.  
21 FEB 78

# LOCATION AND MOVEMENT ANALYSIS SYSTEM LAMAS FLOW

PAGE 75.001

2214  
2215



CALCULATE THE ADJACENT NODE'S TIME VALUES

REF  
PAGE

```
1 . AGAIN THIS IS SIMILAR TO THE TIME ROUTINE FOR THE ROAD NETWORK, AND
2 . AGAIN THE MAIN DIFFERENCE IS THE DATA BASE.
3 .
4 . CALCULATE THE TIME IF TRAVEL BETWEEN J AND THIS NODE
5 . CALCULATE CUMULATIVE TIME FOR THIS LINK
6 . IF THE CUMULATIVE TIME CALCULATED IS LESS THAN THE WORKING LIST ENTRY
7 .   SET NEW CUMULATIVE TIME
8 .   SET NEW TIME MEASURE
9 .   SET PREDECESSOR NODE
10 .  SET CUMULATIVE NORTH
11 .  SET NORTH MEASURE
12 . endif THE CUMULATIVE TIME CALCULATED IS LESS
```

2217  
2218  
2219  
2220  
2221  
2222  
2223  
2224  
2225  
2226  
2227  
2228

TRM, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
LANAS FLOW

PAGE 77

CONSTRUCT SOLUTION<VECTORS AND ROUTE<VECTOR

REF  
PAGE

```
*****
* 1 . THIS ROUTINE IS IDENTICAL TO THE ROAD NETWORK ROUTINE WHICH PERFORMS
* 2 . THE SAME TASK EXCEPT THAT THE WORKING LIST IS REFERENCED IN ORDER TO
* 3 . OBTAIN CERTAIN VALUES, SUCH AS 1) TOTAL WORTH MEASURE, 2) STARTING
* 4 . TIME, 3) TIME MEASURE (LINK TIME), 4) PREDECESSOR'S CUMULATIVE TIME,
* 5 . AND 5) THIS NODE'S CUMULATIVE TIME
*****
2230
2231
2232
2233
2234
*****
```



DISPLAY RESULTS

REF  
PAGE

```
*****
1  do UPON FUNCTION REQUEST
2  do FOREVER
3  DISPLAY FUNCTION MENU
4  WAIT FOR USER INPUT
5  do CASE OF
6  TABLE:
7  PRINT TABLE OF ROUTE NUMBERS WITH ASSOCIATED UNIT NAMES
8  EXIT:
9  return TO CALLING PROGRAM
10 enddo CASE OF
11 enddo FOREVER
12 enddo UPON FUNCTION REQUEST
*****
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
```

TRU, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM

PAGE 79

\*\*\*\*\*  
\*  
\* INDEX TO DATA ITEMS \*  
\*  
\*\*\*\*\*



TRCA, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
INDEX TO DATA ITEMS

PAGE 79.001

INDEX TO DATA ITEMS

PAGE	LINE	TYPE	NAME AND REFERENCES
	DI		MAIN PROGRAM
	26		PREPARATION
	35		LAMAS
	DI		MAP+DIRECTORY
	26		PREPARATION
	27		CREATE NODE+VECTOR AND MAP+DIRECTORY FILES ON DISK
	DI		MAP+NUMBER
	32		CREATE MAP+DIRECTORY ENTRY FOR THIS MAP
	39		ESTABLISH DIRECTORY IN MEMORY AND READ IN NODE+VECTORS
	DI		MAP+NUMBER
	40		READ THIS MAP'S NODES INTO MAIN MEMORY
	41		CHANGE ADJACENT NODE'S MAP+NUMBER TO INDEX
	DI		MAP+NUMBER
	32		CREATE MAP+DIRECTORY ENTRY FOR THIS MAP
	33		CREATE CROSS-COUNTRY, CONCEALMENT, AND DIRECTORY FILES ON DISK
	DI		MAP+NUMBER
	38		INITIALIZE MAP+NUMBERS AND SCREEN EXTREMES
	39		ESTABLISH DIRECTORY IN MEMORY AND READ IN NODE+VECTORS
	DI		MAP+NUMBER
	41		CHANGE ADJACENT NODE'S MAP+NUMBER TO INDEX
	43		ESTABLISH A FORWARD EDGE OF BATTLE AREA

TRU, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
INDEX TO DATA ITEMS

PAGE 79.002

INDEX TO DATA ITEMS

PAGE	LINE	TYPE	NAME AND REFERENCES
	45		PERFORM INTERDICTION OPERATIONS 11
	47		PRINT MAP+NUMBER; NODE+NUMBER OF NODE+VECTOR NEAREST GIVEN COORDINAT 17
	48		CHANGE CONTENTS OF A NODE+VECTOR 7 12
	50		DELETE A NODE FROM THE NODE+VECTOR ARRAY 9
	51		PRINT CONTENTS OF A NODE+VECTOR 2
	60		CALCULATE SECOND BEST PATH 62
	62		CALCULATE BEST NODE AT WHICH TO INTERDICT 76 77
	67		PERFORM TERRAIN INITIALIZATION 10 11
	68		MAP+NUMBER TO BE USED 8
DI	MAP+NUMBERS		
	37		PREPARE FOR ROAD NETWORK PATH CALCULATIONS 7
	39		ESTABLISH DIRECTORY IN MEMORY AND READ IN NODE+VECTORS 13 14
DI	NODE+NUMBER		
	41		CHANGE ADJACENT NODE'S MAP+NUMBER TO INDEX 16
	43		ESTABLISH A FORWARD EDGE OF BATTLE AREA 5 9 14
	46		PERFORM INTERDICTION OPERATIONS 11
	47		PRINT MAP+NUMBER; NODE+NUMBER OF NODE+VECTOR NEAREST GIVEN COORDINAT 17



## INDEX TO DATA ITEMS

PAGE	LINE	TYPE	NAME AND REFERENCES
	48		CHANGE CONTENTS OF A NODE+VECTOR 7 12
	50		DELETE A NODE FROM THE NODE+VECTOR ARRAY 9
	51		PRINT CONTENTS OF A NODE+VECTOR 2
	60		CALCULATE SECOND BEST PATH 62
	62		CALCULATE BEST NODE AT WHICH TO INTERDICT 76 77
DI	NODE+VECTOR		
	26		PREPARATION 2
	27		CREATE NODE+VECTOR AND MAP+DIRECTORY FILES ON DISK 34 35 36 38 44
	35		LAMAS 7
	43		ESTABLISH A FORWARD EDGE OF BATTLE AREA 14 15
	44		FIND NODE+VECTOR NEAREST TO UTM COORDINATES 9 12 15
	46		PERFORM INTERDICTION OPERATIONS 11 13 15 17
	47		PRINT MAP+NUMBER, NODE+NUMBER OF NODE+VECTOR NEAREST GIVEN COORDINAT 7 11 13 17
	48		CHANGE CONTENTS OF A NODE+VECTOR 8 11 13 16 17 38 39 40
	49		CHANGE CHARACTERISTIC 7
	50		DELETE A NODE FROM THE NODE+VECTOR ARRAY 9
	51		PRINT CONTENTS OF A NODE+VECTOR 7

TRU, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
INDEX TO DATA ITEMS

PAGE 79.004

INDEX TO DATA ITEMS

PAGE	LINE	TYPE	NAME AND REFERENCES
	53		CALCULATE PATH
	78		83
	54		PURGE ALL ROUTES
	13		15
	55		REINITIALIZE 'WORKING LIST' NODE+VECTOR ENTRIES
	4		14
	56		CALCULATE AND ASSIGN TIME AND NORTH VALUES
	33		35
	59		COMPUTE SOLUTION+VECTORS AND ROUTE+VECTOR
	46		58
	60		CALCULATE SECOND BEST PATH
	42		57
	61		FIND THE ORDER OF PRIORITIES FOR BEST NETWORK TRAVEL
	49		
	62		CALCULATE BEST NODE AT WHICH TO INTERDICT
	68		71
DI		NODE+VECTORS	
	27		CREATE NODE+VECTOR AND MAP+DIRECTORY FILES ON DISK
	20		
	33		INITIALIZE MAP+NUMBERS AND SCREEN EXTREMES
	2		
	43		CHANGE CONTENTS OF A NODE+VECTOR
	20		
DI		NUMBER+OF+MAPS	
	32		CREATE MAP+DIRECTORY ENTRY FOR THIS MAP
	9		
DI		ROAD+NETWORK	
	35		LANAS
	6		



LOCATION AND MOVEMENT ANALYSIS SYSTEM  
INDEX TO DATA ITEMS

TRU, INC.  
21 FEB 78

INDEX TO DATA ITEMS

PAGE	LINE	TYPE	NAME AND REFERENCES
			-----
		DI	ROUTE<VECTOR
	53		CALCULATE PATH
	15		
	54		PURGE ALL ROUTES
	6		8
	60		CALCULATE SECOND BEST PATH
	35		59
	61		FIND THE ORDER OF PRIORITIES FOR BEST NETWORK TRAVEL
	53		69
	62		CALCULATE BEST NODE AT WHICH TO INTERDICT
	41		70
	72		CALCULATE CROSS-COUNTRY PATH
	24		
		DI	ROUTE<VECTORS
	51		FIND THE ORDER OF PRIORITIES FOR BEST NETWORK TRAVEL
	66		
	62		CALCULATE BEST NODE AT WHICH TO INTERDICT
	74		
		DI	SOLUTION<VECTOR
	54		PURGE ALL ROUTES
	9		11
	14		
	56		CALCULATE AND ASSIGN TIME AND NORTH VALUES
	40		42
	44		46
	49		67
	59		COMPUTE SOLUTION<VECTORS AND ROUTE<VECTOR
	24		25
	46		47
	48		53
	54		58
	61		FIND THE ORDER OF PRIORITIES FOR BEST NETWORK TRAVEL
	49		
	62		CALCULATE BEST NODE AT WHICH TO INTERDICT
	48		
	72		CALCULATE CROSS-COUNTRY PATH
	27		

TRM, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
INDEX TO DATA ITEMS

PAGE 79.006

INDEX TO DATA ITEMS

PAGE	LINE	TYPE	NAME AND REFERENCES
		DI	SOLUTION+VECTORS
	53		CALCULATE PATH
	15		
	59		COMPUTE SOLUTION+VECTORS AND ROUTE+VECTOR
	33		43 65
	60		CALCULATE SECOND BEST PATH
	35		59
	61		FIND THE ORDER OF PRIORITIES FOR BEST NETWORK TRAVEL
	53		66 69
	62		CALCULATE BEST NODE AT WHICH TO INTERDICT
	41		70 73 74
	72		CALCULATE CROSS-COUNTRY PATH
	24		29
		DI	TERRAIN+MODEL
	35		LAMAS
	8		
		DI	UNIT+COUNTER
	42		INITIALIZE UNIT+VECTORS
	9		14 26 58
	69		ESTABLISH UNIT+VECTORS
	11		14 48
		DI	UNIT+VECTIR
	42		INITIALIZE UNIT+VECTORS
	27		33 34 40 46 52
	52		CHANGE PROPERTIES OF A UNIT
	10		
	60		CALCULATE SECOND BEST PATH
	60		61
	69		ESTABLISH UNIT+VECTORS
	35		



TOM, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
INDEX TO DATA ITEMS

PAGE 79.007

INDEX TO DATA ITEMS

PAGE	LINE	TYPE	NAME AND REFERENCES
		DI	UNIT+VECTORS
	37		PREPARE FOR ROAD NETWORK PATH CALCULATIONS
	42		INITIALIZE UNIT+VECTORS
	60		CALCULATE SECOND BEST PATH
	62		CALCULATE BEST NODE AT WHICH TO INTERDICT
	67		PERFORM TERRAIN INITIALIZATION
	69		ESTABLISH UNIT+VECTORS

LOCATION AND MOVEMENT ANALYSIS SYSTEM

TRIG, INC.  
21 FEB 78

\*\*\*\*\*  
\*  
\* INDEX TO FLOW SEGMENTS \*  
\*  
\*\*\*\*\*



TRM, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
INDEX TO FLOW SEGMENTS

PAGE 80.001

INDEX TO FLOW SEGMENTS

PAGE	LINE	TYPE	NAME AND REFERENCES
56		FS	CALCULATE AND ASSIGN TIME AND WORTH VALUES 53 CALCULATE PATH 08
62		FS	CALCULATE BEST NODE AT WHICH TO INTERDICT 45 PATH DETERMINATION AND DISPLAY 32
72		FS	CALCULATE CROSS-COUNTRY PATH 71 PERFORM TERRAIN PATH CALCULATIONS 7
53		FS	CALCULATE PATH 45 PATH DETERMINATION AND DISPLAY 26
60		FS	CALCULATE SECOND BEST PATH 45 PATH DETERMINATION AND DISPLAY 28
76		FS	CALCULATE THE ADJACENT NODE'S TIME VALUES 75 CALCULATE THE CROSS-COUNTRY PATH 16
75		FS	CALCULATE THE CROSS-COUNTRY PATH 72 CALCULATE CROSS-COUNTRY PATH 19
41		FS	CHANGE ADJACENT NODE'S MAP+NUMBER TO INDEX 38 INITIALIZE MAP+NUMBERS AND SCREEN EXTREMES 6

TRM, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
INDEX TO FLOW SEGMENTS

PAGE 80.002

INDEX TO FLOW SEGMENTS

PAGE	LINE	TYPE	NAME AND REFERENCES
49		FS	CHANGE CHARACTERISTIC 48 CHANGE CONTENTS OF A NODE+VECTOR 25 27 29 31 33 35
48		FS	CHANGE CONTENTS OF A NODE+VECTOR 46 PERFORM INTERDICTION OPERATIONS 13
70		FS	CHANGE ENTERED COORDINATES TO AN INDEX INTO THE DATA BASE 69 ESTABLISH UNIT+VECTORS 34
52		FS	CHANGE PROPERTIES OF A UNIT 46 PERFORM INTERDICTION OPERATIONS 19
57		FS	COMPUTE NEXT NODE TO LABEL 53 CALCULATE PATH 12
59		FS	COMPUTE SOLUTION+VECTORS AND ROUTE+VECTOR 53 CALCULATE PATH 15 60 CALCULATE SECOND BEST PATH 35 59 61 FIND THE ORDER OF PRIORITIES FOR BEST NETWORK TRAVEL 69 62 CALCULATE BEST NODE AT WHICH TO INTERDICT 41 70 73
77		FS	CONSTRUCT SOLUTION+VECTORS AND ROUTE+VECTOR 72 CALCULATE CROSS-COUNTRY PATH 24



TRM, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
INDEX TO FLOW SEGMENTS

PAGE 80.003

INDEX TO FLOW SEGMENTS

PAGE	LINE	TYPE	NAME AND REFERENCES
33		FS	CREATE CROSS-COUNTRY, CONCEALMENT, AND DIRECTORY FILES ON DISK 26 PREPARATION 3
30		FS	CREATE LINK CARD 27 CREATE NODE+VECTOR AND MAP+DIRECTORY FILES ON DISK 10
32		FS	CREATE MAP+DIRECTORY ENTRY FOR THIS MAP 27 CREATE NODE+VECTOR AND MAP+DIRECTORY FILES ON DISK 43
28		FS	CREATE NODE CARD 27 CREATE NODE+VECTOR AND MAP+DIRECTORY FILES ON DISK 7
27		FS	CREATE NODE+VECTOR AND MAP+DIRECTORY FILES ON DISK 26 PREPARATION 2
50		FS	DELETE A NODE FROM THE NODE+VECTOR ARRAY 46 PERFORM INTERDICTION OPERATIONS 15
78		FS	DISPLAY RESULTS 71 PERFORM TERRAIN PATH CALCULATIONS 9
43		FS	ESTABLISH A FORWARD EDGE OF BATTLE AREA 37 PREPARE FOR ROAD NETWORK PATH CALCULATIONS 11

TR4, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
INDEX TO FLOW SEGMENTS

PAGE 50.004

INDEX TO FLOW SEGMENTS

PAGE	LINE	TYPE	NAME AND REFERENCES
39		FS	ESTABLISH DIRECTORY IN MEMORY AND READ IN NODE+VECTORS 38 INITIALIZE MAP+NUMBERS AND SCREEN EXTREMES 2
69		FS	ESTABLISH UNIT+VECTORS 67 PERFORM TERRAIN INITIALIZATION 13
44		FS	FIND NODE+VECTOR NEAREST TO UTM COORDINATES
61		FS	FIND THE ORDER OF PRIORITIES FOR BEST NETWORK TRAVEL 45 PATH DETERMINATION AND DISPLAY 30
66		FS	IMPLEMENT ALGORITHMS AND FUNCTIONS USING THE CROSS-COUNTRY DATA BASE 35 LAMAS 9
36		FS	IMPLEMENT ALGORITHMS AND FUNCTIONS USING THE NODE+VECTOR DATA BASE 35 LAMAS 7
38		FS	INITIALIZE MAP+NUMBERS AND SCREEN EXTREMES 37 PREPARE FOR ROAD NETWORK PATH CALCULATIONS 7
42		FS	INITIALIZE UNIT+VECTORS 37 PREPARE FOR ROAD NETWORK PATH CALCULATIONS 9
35		FS	LAMAS



INDEX TO FLINK SEGMENTS

PAGE	LINE	TYPE	NAME AND REFERENCES
63		FS	MAP+NUMBER TO BE USED 67 PERFORM TERRAIN INITIALIZATION 11
74		FS	OBTAIN THE PATH PARAMETERS 72 CALCULATE CROSS-COUNTRY PATH 15
73		FS	OBTAIN UNIT NAMES TO BE CONSIDERED 72 CALCULATE CROSS-COUNTRY PATH 13
45		FS	PATH DETERMINATION AND DISPLAY 36 IMPLEMENT ALGORITHMS AND FUNCTIONS USING THE NODE+VECTOR DATA BASE 9
46		FS	PERFORM INTERDICTIVE OPERATIONS 45 PATH DETERMINATION AND DISPLAY 24
67		FS	PERFORM TERRAIN INITIALIZATION 66 IMPLEMENT ALGORITHMS AND FUNCTIONS USING THE CROSS-COUNTRY DATA BASE 11
71		FS	PERFORM TERRAIN PATH CALCULATIONS 66 IMPLEMENT ALGORITHMS AND FUNCTIONS USING THE CROSS-COUNTRY DATA BASE 13
26		FS	PREPARATION
37		FS	PREPARE FOR ROAD NETWORK PATH CALCULATIONS 36 IMPLEMENT ALGORITHMS AND FUNCTIONS USING THE NODE+VECTOR DATA BASE 7

TRM, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
INDEX TO FLOW SEGMENTS

PAGE 80.006

INDEX TO FLOW SEGMENTS

PAGE	LINE	TYPE	NAME AND REFERENCES
63		FS	PRESENT RESULTS 43 PATH DETERMINATION AND DISPLAY 34
51		FS	PRINT CONTENTS OF A NODE+VECTOR 46 PERFORM INTERDICTION OPERATIONS 17
47		FS	PRINT MAP+NUMBER, NODE+NUMBER OF NODE+VECTOR NEAREST GIVEN COORDINATES 46 PERFORM INTERDICTION OPERATIONS 11
65		FS	PRINT PATH STATISTICS 63 PRESENT RESULTS 12
64		FS	PRINT TABLE OF ROUTE NUMBERS WITH ASSOCIATED UNIT NAMES 63 PRESENT RESULTS 10 78 DISPLAY RESULTS 7
54		FS	PURGE ALL ROUTES 53 CALCULATE PATH 26 50 CALCULATE SECOND BEST PATH 10 61 FIND THE ORDER OF PRIORITIES FOR BEST NETWORK TRAVEL 13 62 CALCULATE BEST NODE AT WHICH TO INTERDICTION 16 72 CALCULATE CROSS-COUNTRY PATH 12



TRW, INC.  
21 FEB 73

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
INDEX TO FLOW SEGMENTS

PAGE 00.007

INDEX TO FLOW SEGMENTS

PAGE	LINE	TYPE	NAME AND REFERENCES
40		FS	READ THIS MAP'S NODES INTO MAIN MEMORY
	39		ESTABLISH DIRECTORY IN MEMORY AND READ IN NODE+VECTORS
			26
55		FS	REINITIALIZE 'WORKING LIST' NODE+VECTOR ENTRIES
	53		CALCULATE PATH
			78 83
	60		CALCULATE SECOND BEST PATH
			42 57

TRC, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM

PAGE 81

\*\*\*\*\*  
\*  
\* SEGMENT REFERENCE TREES \*  
\*  
\*\*\*\*\*



TRW, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
SEGMENT REFERENCE TREES

PAGE 81.001

PREPARATION

LN	DEF	SEGMENT
1	26	PREPARATION
2	27	CREATE NODE+VECTOR AND MAP+DIRECTORY FILES ON DISK
3	28	CREATE NODE CARD
4	30	CREATE LINK CARD
5	32	CREATE MAP+DIRECTORY ENTRY FOR THIS MAP
6	33	CREATE CROSS-COUNTRY, CONCEALMENT, AND DIRECTORY FILES ON DISK

LHMAS

LN	DEF	SEGMENT
1	35	LHMAS
2	36	IMPLEMENT ALGORITHMS AND FUNCTIONS USING THE NODE+VECTOR DATA BASE
3	37	PREPARE FOR ROAD NETWORK PATH CALCULATIONS
4	38	INITIALIZE MAP+NUMBERS AND SCREEN EXTREMES
5	39	ESTABLISH DIRECTORY IN MEMORY AND READ IN NODE+VECTORS
6	40	READ THIS MAP'S NODES INTO MAIN MEMORY
7	41	CHANGE ADJACENT NODE'S MAP+NUMBER TO INDEX
8	42	INITIALIZE UNIT+VECTORS
9	43	ESTABLISH A FORWARD EDGE OF BATTLE AREA
10	45	PATH DETERMINATION AND DISPLAY
11	46	PERFORM INTERDICTIVE OPERATIONS
12	47	PRINT MAP+NUMBER, NODE+NUMBER OF NODE+VECTOR NEAREST GIVEN COORDINATES
13	48	CHANGE CONTENTS OF A NODE+VECTOR
14	49	CHANGE CHARACTERISTIC
15	50	DELETE A NODE FROM THE NODE+VECTOR ARRAY
16	51	PRINT CONTENTS OF A NODE+VECTOR
17	52	CHANGE PROPERTIES OF A UNIT
18	53	CALCULATE PATH
19	54	PURGE ALL ROUTES
20	55	REINITIALIZE 'WORKING LIST' NODE+VECTOR ENTRIES
21	56	CALCULATE AND ASSIGN TIME AND WORTH VALUES
22	57	COMPUTE NEXT NODE TO LABEL
23	59	COMPUTE SOLUTION+VECTORS AND ROUTE+VECTOR
24	60	CALCULATE SECOND BEST PATH
25	54	PURGE ALL ROUTES
26	59	COMPUTE SOLUTION+VECTORS AND ROUTE+VECTOR
27	55	REINITIALIZE 'WORKING LIST' NODE+VECTOR ENTRIES
28	61	FIND THE ORDER OF PRIORITIES FOR BEST NETWORK TRAVEL
29	54	PURGE ALL ROUTES
30	59	COMPUTE SOLUTION+VECTORS AND ROUTE+VECTOR
31	62	CALCULATE BEST NODE AT WHICH TO INTERDICT
32	54	PURGE ALL ROUTES
33	59	COMPUTE SOLUTION+VECTORS AND ROUTE+VECTOR
34	63	PRESENT RESULTS



TRM, INC.  
21 FEB 78

LOCATION AND MOVEMENT ANALYSIS SYSTEM  
SEGMENT REFERENCE TRELS

PAGE 81.003

LOWAS (CONTINUED)

LN	DEF	SEGMENT
35	64	PRINT TABLE OF ROUTE NUMBERS WITH ASSOCIATED UNIT NAMES
36	65	PRINT PATH STATISTICS
37	66	IMPLEMENT ALGORITHMS AND FUNCTIONS USING THE CROSS-COUNTRY DATA BASE
38	67	PERFORM TERRAIN INITIALIZATION
39	68	HAP+NUMBER TO BE USED
40	69	ESTABLISH UNIT+VECTORS
41	70	CHANGE ENTERED COORDINATES TO AN INDEX INTO THE DATA BASE
42	71	PERFORM TERRAIN PATH CALCULATIONS
43	72	CALCULATE CROSS-COUNTRY PATH
44	73	PURGE ALL ROUTES
45	74	OBTAIN UNIT NAMES TO BE CONSIDERED
46	75	OBTAIN THE PATH PARAMETERS
47	76	CALCULATE THE CROSS-COUNTRY PATH
48	77	CALCULATE THE ADJACENT NODE'S TIME VALUES
49	78	CONSTRUCT SOLUTION+VECTORS AND ROUTE+VECTOR
50	79	DISPLAY RESULTS
51	80	PRINT TABLE OF ROUTE NUMBERS WITH ASSOCIATED UNIT NAMES